شبكات معلومات الاقراص المكتنزة CD-ROM التجربة العراقية والتعاون العربي

عامر ابراهيم قنديلجي

EMISTRY	DE
	M
	DI
YSICS	M
	SC
	PE
	EI
	BA
	SA
OLOGY	PE
	LI
	D A
ATHEMATICS	RI
	M
OMPUTER STEM	P
	D
UTOMATIC OGRAMING	
	PR A
ANFACTURE	
ILITARY	I

TUBLE 2:- EXI

مكونات النظام :- يتكون النظام اساساً من اربعة ملفات هي :

١- Main .pro:- وهو الملف الرئيسي ومهمته استدعاء الملفات الثلاثة الباقية وذلك بناء على مايدخله المستعمل .

٢- Dignos .pro :- ويقوم هذا الملف بعملية تشخيص حالة المريض .

٣- Treat .pro :- ومهمته تحديد الاسلوب الامثل للعلاج من خلال تحديد في اي مرحلة من السرطان تقع حالة المريض .

٤- Predic .pro :- ويحدد احتمالية اصابة المستخدم بالمرض في المستقبل .

مكونات قاعدة المعرفة

والان سنلقي نظرة على قواعد الانتاج التي تمثل طلب قاعدة المعرفة للنظام. وهذه القواعد يتم كتابتها بلغة انكليزية مبسطة في الشكل المعتاد IF THEN ELES حتى يتمكن القارئ من أخذ فكرة عن عمل النظام ويجب التنويه ان هذه القواعد لاتحتوي على كلا من واجهة المستفيد ولا على النوافذ التي يحويها النظام وانما تحوي المعلومات الطبية بشكل مبسط . وكما هو معروف فان النظام يحوي على تشخيص وعلاج والتنبؤ بالمرض وسيتم عرض عينات من قواعد الانتاج الخاصة بكل عملية .

١- Dignosis:- من خلال عملية التشخيص يمكننا معرفة ما اذا كان المريض مصاب بالسرطان ايضا فان التشخيص يساعد على معرفة ما اذا كانت حالة المريض الصحية تسمح باجراء عملية طبية اما لا كما يساعد التشخيص على معرفة ما اذا كان هناك امل في شفاء المريض ويتقسم التشخيص الى ثلاث اقسام رئيسية وهي :

Clinical History , Respiratory Signs & Symptoms Medical Investigations

Clinical History :- cunsitutes about 5 % of the diagnosis prosiss

Rule 1:-

IF PATIENT IS MALE :-

THEN POSSIBILITY OF CATCHING THE DISEASE DUE TO SEX IS 9%

ELSF POSSIBILITY OF CATCHING THE DISEASE DUE TO SEX IS 4%

****

Rule 2:-
IF PATIENT AGT . IS >=40 AND <=70
THEN POSSIBLLTTY OF CATCHNG THE DISEASE DUE TO AGE IS 9%
ELSE POSSIBLLTTY OF CATCHNG THE DISEASE DUE TO AGE IS 2%
****

٢- SIGNS & SYMPTOMS :- وتكون حوالي 25٪ من عملية التشخيص

وتقسم لجزئين مما الـRESPIRATORY SIGNS (15%) والـ NON
(10%) RESPIRATORY SIGNS

RULE 14 :-
IF
(11) NON RESPIRATORY SIGNS .
RULE 25 :-
IF PATIENT CETS FATTUE EASY
THEN POSSIBLLTIY OF CATCHING THE DISEASE DUE TO THE FATIGUNESS FACTOR IS 10 %
ELSE POSSIBILRRY OF CATCHING THE DISEASE DUE TO THE FATIGUNESS FACTOR IS 0 %
****

٣- MEDICAL INVESTIONS:- وهي تكون 75٪ من عملية التشخيص

ونصل صلب عملية التشخيص.

RULE 34 :-
IF PATIENT X-RAY SHOWED ANY ABNORMAL OPACITY THEN PATIENT IS A LUNG CANCER PATENT .
****

2- TREATMENT:-وبطبيعة المثال فان الغرض من هذه العملية هـو تحديد في مرحلة السرطان وذلك حتى يتم معرفة الطريقة المثلى للعلاج ويتم تحديد هذه المرحلة بواسطة ما يسمى بالـ STAGING FACTOR الذي يحدد بدوره بواسطة عدة عوامل مرتبطة بحجم الورم وامتداداته والامراض الاخرى .

```
RULE 89 :-
IF THE SIZE OF THE TUMOR IS LARGE :-
THEN SET THE STAGING FACTOR -1 TO 30
ELSE IF THE SIZE OF THE TUMOR IS MEDIUM:-
THEN SET THE STAGING FACTOR -1 TO 20
ELES SET THE STAGING FACTOR-1 TO 10
```

****

3- PREDICTION:-والغرض من هذه العملية هو التنبؤ باحتمالية الاصابة بالمرض مستقبلا ويتم تحديد هذه العملية عن طريق عوامل الوراثة والتدخين والبيئة وظروف العمل .

```
RULE 53:-
IF THE CLIENT IS MALE :-
THEN THE POSSIBILITY OF BEING INFECTED DUE TO SEX IS 40%
THEN THE POSSIBILITY OF BEING INFECTED DUE TO SEX IS 10%
```

****

```
RULE 54 :-
IF THE CLIENT AGE IS BETWEEN 40 AND 70 :-
THEN THE POSSIBILITY OF BEING INFECTED DUE TO AGE IS 30%
ELES THE POSSIBILITY OF BEING INFECTED DUE TO AGE IS 10 %
```

****

شبكات معلومات الاقراص المكتنزة CD-ROM

التجربة العراقية والتعاون العربي

عامر ابراهيم قنديلجي

مقدمة ومستخلص

لقد أصبحت تقنية الأقراص المكتنزة وقواعدها المخزونة عليها أمراً لا مفر منه للباحثين والمستفيدين من خدمات المعلومات المتطورة التي تقدمها المكتبات ومراكز المعلومات وتشهد العديد من مناطق العالم ، ومنها الدول النامية تحولا ملحوظاً نحو استثمار قواعد الأقراص واستخدامها عبر شبكات معلومات وطنية واقليمية مناسبة ، كذلك فقد خطت المكتبات ومراكز المعلومات في العراق والاردن خطوات مهمة على طريق استخدام هذه الاقراص الا ان هذ التجارب تحتاج الى مزيد من العناية والتثمين والتنسيق والتعاون . ومن هنا تتوضح اهداف البحث وأهميته .

وقد حاول الباحث الاستفادة من الدراسات الوثائقية والميدانية السابقة في تحليل واستقراء المعومات الواردة فيها . وقسم البحث الى محاور ثلاث اساسية هي :

أولاً : ما هية الأقراص المكتنزة وأنواعها وتطور استخدامها، وخاصة ما له علاقة باقراص اقرأ ما في الذاكرة (CD-ROM) . ثم انتقل الباحث . وبشكل اكثر تركيزاً وايجازاً الى مفهوم بنوك وشبكات المعلومات المعتمدة على الاقراص .

ثانياً : استعراض مركز وموجز لبعض التجارب العالمية الاقليمية . والعربية (الاردنية) والمحلية (العراقية) في مجال استخدام قواعد الأقراص .

ثالثاً : الاستنتاجات والتوصيات ، من حيث اهمية استخدام قواعد الأقراص البيليوغرافية (Bibliographic Databases) على وجه التخصيص وكذلك بعض السمات الايجابية والسلبية للتجربة العراقية في بناء شبكة الأقراص التعاونية ، وتجربة جامعة البرموك في الاردن وتجربتها عبر شبكة (LAN). وقد ركز الباحث في توصياته ومقترحاته على تنظيم خطط

التعاون المستقبلية في بناء شبكات معلومات الأقراص المكتنزة ، من خلال مراحل أربعة اساسية ، على المستويين القطري المحلي لكل من العراق والاردن، المستوى القومي ، مع امكانية دعم المنظمات والاتحادات العربية الأقليمية لمثل هذا التحرك والتعاون .

كذلك فقد اشتمل البحث على عدد محدد من المخططات والملاحق الضرورية والمناسبة .

أولا :

الأقراص المكتنزة

(Compact Disks / CD)

يختلف العديد من الكتاب والباحثين العرب في تعريب مصطلح (Compact Disk) بأنواعه واستخداماته المختلفة ، فمنهم من أطلق عليه اسم (الأقراص المضغوطة) ، ومنهم من ترجمه (الأقراص المتراصة) ، ثم (اقراص الليزر او المليزرة)، والاقراص المرئية أو البصرية . أما الكاتب فقد التزم بمصطلح (الأقراص المكتنزة) منذ اول دراسة نشرت له في نهاية عقد الثمانينات حول الموضوع . وقد حاول نجيب الشربجي معالجة هذا الموضوع بشكل مسهب . [١] .

ومن المفيد أن نؤكد هنا الى أن الاقراص المكتنزة (بأنواعها المختلفة) عبارة عن وسيلة مستحدثة ومتطورة ، تعمل بتقنية أشعة الليزر ، وتستثمر في مجال تخزين كميات هائلة من البيانات والمعلومات المقروءة أو المسموعة أو المرئية ، وعلى قرص او اسطوانه قطرها ٢٥ر٥ أو ٨ بوصة . سهلة الاستخدام والتداول ، وتستخدم هذه الاقراص في نقل وبث واسترجاع المعلومات للمستفيدين بسرعة كبيرة وكفاءة عالية جداً ، مقارنة بالوسائط

(١) نجيب الشربجي . الاقراص المتراصة : تفاعلات مجلة رسالة المكتبة . مج ٢٨ ، ع ١ ٢ ، كانون اول ١٩٩٣ ، ص

الاخرى . أما مجالات الاقراص المكتنزة فمتنوعة بتنوع أنواعها ، فبعضها يستخدم في المجالات العلمية والبحثية . واخرى في المجالات الاعلامية والترفيهية ، وكذلك في مجالات الوثائق والتوثيق والمكتبات [٢] .

وعلى هذا الاساس فان هذه الاقراص تمثل قمة التطور التقني في تخزين وتناقل المعلومات . حتى ان البعض من الكتاب يعادلها باختراع كوتنبرغ للطباعة ، فهي تتفوق على أنواع التقنيات الاخرى المعروفة والمستخدمة . كالاقراص والاشرطة المغنطة بجوانب عدة أهمها : [٣] .

١- ان البيانات المخزونة على الأقراص والاشرطة المغنطة معرضة للتلف والفقدان ، لأن التسجيلات المغنطة يمكن ان تتأثر او تتلاشى عند تعرضها لمجال مغناطيسي مؤثر ، اضافة الى ان مسارات (Tracks) الاقراص المغنطة معرضة للأتربة والغبار لكون سطح القرص غير محمي بطبقة عازلة خارجية ، كما هو الحال مع الأقراص المكتنزة . حيث ان هذه الاخيرة تحمي وتحصن البيانات والمعلومات المخزونة عليها من التلف والتلاشي بوجود مادة عازلة تكسو السطح الخارجي للقرص وتعمل كغطاء واقي وحافظ للبيانات المنتشرة في المسارات الموجودة على سطح القرص.

٢- تعمل الأقراص المكتنزة بواسطة تقنية وأشعة الليزر المتمثلة بحزم ضوئية توجه الى سطح القرص ، ولها القدرة على اختراق الحاجز والمادة العازلة لتصل الى المواقع المتناهية الصغر في سطح القرص ، بغرض قراءة أو تسجيل البيانات ، من دون الحاجة الى ملامسة فعلية لسطح القرص ، كما هي الحال في الأقراص المغنطة .

ويمكننا القول بأن الاقراص المكتنزة غير قابلة للتلف من جراء الاستخدام

---

(٢) عامر ابراهيم قنديلجي . استخدام الحواسيب في خدمات المعلومات : ملامح عن التجربة العراقية . المجلة العربية للمعلومات . مج ١٥، ع١، ١٩٩٤ . ص ٧٠ - ٧٢ .

(٣) ايمان فاضل السامرائي . الأوعية المتعددة وتطور الأقراص منذ عام ١٩٧٧ وحتى عام ١٩٩٢ . المجلة العربية للمعلومات . مج ١٥،ع١ ، ١٩٩٤ ، ص ٩٦ - ٩٩ .

والتداول ، ويمكن المحافظة على كفاءة وجودة ما تخزنه من بيانات ومعلومات - مقروءة أو مسموعة او مرئية - الى ما لا نهاية .

٣- كذلك فقد حققت تقنية وأشعة الليزر غاية أخرى هي تقليص حجم القرص المستخدم ، مع زيادة هائلة في كمية البيانات المخزونة ، حيث تفوق الامكانات التخزينية للقرص المكتنز مائة مرة القدرة التخزينية للقرص المغنط.

٤- ان تكلفة تخزين البيانات والمعلومات على الاقراص المكتنزة هي أقل بكثير من تكلفة التخزين على الاقراص المغنطة . حيث تقدر قيمة تخزين وثيقة قياس ١١× ٥ر٨ بوصة على قرص مكتنز (٤) سنتات فقط، بينما تكلف مثل هذه الوثيقة (٢٥ر١) دولار على القرص المغنط .

أقراص اقرأ ما في الذاكرة CD-ROM .

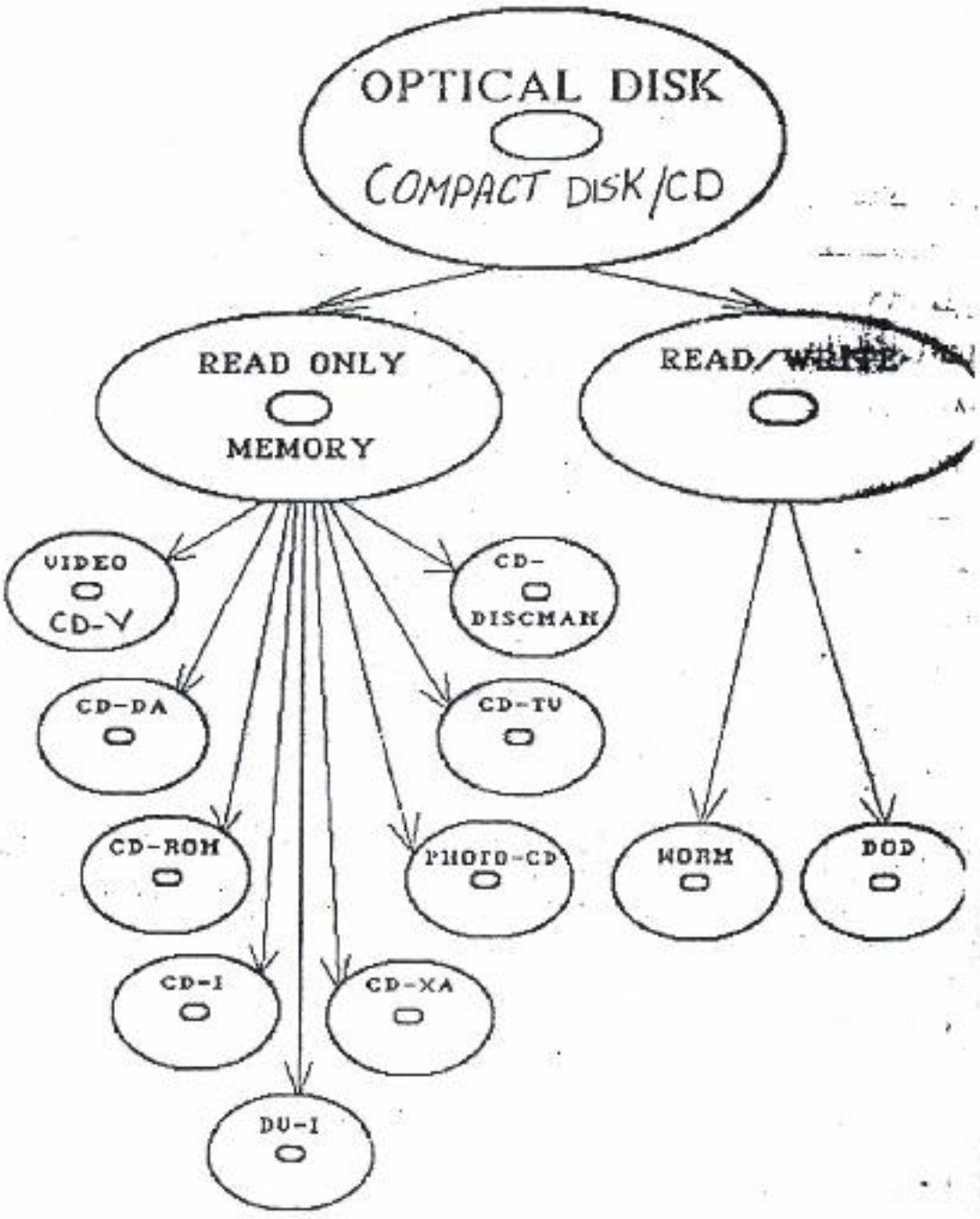
تقسم الاقراص المكتنزة (المرئية) الى قسمين اساسين هما :

أ . اقراص اقرأ ما في الذاكرة (Read Only Memory) وتأتي من المجهزين جاهزة ، لا يمكن تحديثها او تغييرها [١٤]

ب - اقراص يمكن الاضافة عليها وتسمى (Read / write) وقابلة للمسح (وكما هو مبين في الشكل المرفق رقم -١)

---

١٤١ نفس المصدر السابق ، ص ٩٩



مخطط رقم (١)

الأنواع المختلفة للأقراص المكتنزة

تطور استخدام قواعد الأقراص

وعموماً فان القرص المكتنز من نوع اقرأ ما في الذاكرة فقط المستخدم في المكتبات ومراكز التوثيق والمعلومات ،ومن القياسات الاكثر شيوعاً (٥ر٢٥ بوصة) يختزن من البيانات ما يصل الى (٦٦٠) مليون رمز (660 Megabyte) وتعادل هذه الكمية من البيانات ما مقداره (٣٣٠ر٠٠٠) صفحة ورقية مطبوعة ، وكذلك ما يعادل البيانات المخزونة على (١٨٠٠) قرص مرن(Floppy Disk) [٥].

وعلى هذا الاساس فقد اصبحت اقراص اقرأ ما في الذاكرة وسيطاً مهماً ومناسباً في التعامل مع المعلومات ، خزناً ومعالجة واسترجاعاً . فقد بينت الدراسات والتقارير حول هذا الموضوع أن نمواً متزايداً في عدد المستخدمين قد طرأ لهذا النوع من التقنيات . فقد بلغ عدد اجهزة تشغيل الاقراص (CD - ROM Drive) في السنوات الاولى لظهور هذه التقنية ، ما مجموعه (٢٥٠ر٠٠٠) جهاز في العالم ، وذلك عام ١٩٨٨ . ونستطيع تصور الزيادة التي حدثت بين هذا التاريخ ، خاصة بعد اطلاعنا على النمو والزيادة الكبيرة في عدد الاقراص والقواعد المنتجة بعد ذلك التاريخ (انظر الجدول رقم ١٠) . وفي تقرير حول عدد المكتبات التي تستخدم ، او تخطط للاستخدام ، لاقراص اقرأ ما في الذاكرة في المملكة المتحدة ، حيث بلغت ٦٠٪ من مجموع المكتبات فيها ، في عام ١٩٨٩ [٦].

كذلك فقد ازدادت أهمية اقراص (CD-ROM) واستخداماتها في المكتبات ومراكز المعلومات بتطور استثمار شبكات المعلومات متعددة

---

(5) Silver platter, 1993 Directory of CD-ROM Databases Northwood MA. Silver Platter information , 1993 . P.4 .
(6) John Akeroyd. CD- ROM networkes. The Electronic Library. Vol.9. No. 1, Feb. 1991 . p.21 .

المستفيدين(Multi-users) سواء كانت شبكات محلية (LAN) او شبكات اخرى متباعدة .

وتطور التعامل مع قواعد الاقراص من خلال الاعداد المتزايدة من الشركات المنتجة لها ، والذين تزايد عددهم من (٤٨) شركة ومجهز عام ١٩٨٦، الى (٢٢١٢) في عام ١٩٩١ [٧].

وفي دلالات اخرى على استخدام قواعد الأقراص المكتنزة اقرأ ما في الذاكرة مقارنة بالتقنيات والوسائل الاخرى المستخدمة في البحث . فقد ارتفع عدد عناوين وقواعد الاقراص من قرصين اثنين عام ١٩٨٥ الى (٦٤٣٨) عنواناً او قاعدة اقراص في عام ١٩٩٢. وذلك مقارنة بعدد قواعد البحث بالاتصال المباشر (Online Searching)والتي كانت قد ارتفعت من (٢٤٥٣) في عام ١٩٨٥ ، الى (٥٤٠٣) قاعدة عام ١٩٩٢ . وبنسبة مئوية هي (٢٢٠٪) مقارنة بنسبة (٣٢١٩٠٠٪) مثلتها زيادة الأقراص للسنوات المبينة أعلاه ، وبعبارة أخرى فإن تطور استخدامات الاقراص المكتنزة وقواعدها المختلفة كان هائلا مقارنة بتطور استخدامات البحث بالاتصال المباشر . ويبين الجدول رقم (١) في ادناه التطورات السنوية لكل من الاقراص والبحث بالاتصال المباشر [٨] .

السنة	قواعد الأقراص CD-ROM	نسبة الزيادة	قواعد البحث الاتصال المباشر Online	نسبة الزيادة
١٩٨٥	٢	-	٢٤٥٣	-
١٩٨٦	٦٢	٣١٠٠٪	٢٩٠١	١١٨٪
١٩٨٧	٨٣٥	٤١٧٥٠٪	٣٣٦٩	١٣٧٪
١٩٨٨	٣٢٨	١٦٤٠٠٪	٣٦٩٩	١٥٠٪
١٩٨٩	٧٥٣	٣٧٦٥٠٪	٤٠٦٢	١٦٥٪
١٩٩٠	١٦٩٤	٨٤٧٠٠٪	٤٤٦٥	١٨٢٪
١٩٩١	٣٣٨٩	١٦٩٤٥٠٪	٤٩١٢	٢٠٠٪
١٩٩٢	٦٤٣٨	٣٢١٩٠٠٪	٥٤٠٣	٢٢٠٪

جدول رقم (١)
تطور قواعد الاقراص وقواعد البحث بالاتصال المباشر

(٧) ايان فاضل السماراني . مصدر سابق . ص ١٢٢ .

(٨) عامر ابراهيم قنديلجي . بناء شبكة وطنية للمكتبات ومراكز المعلومات في العراق : المشاكل والحلول المقترحة . ندوة نظم المعلومات في القرن الحادي والعشرين . جامعة السرمك ٢٨- ٣٠/ ٦/ ١٩٩٣ . ص ١١

الانواع المستخدمة من قواعد الاقراص اقرا ما في الذاكرة

هنالك اربعة انواع مستخدمة في خدمات المعلومات المسترجعة من الاقراص المكتنزة (CD-ROM) والتي يمثل النوع الاول الاكثر استخداماً وشيوعاً في العراق والاردن وهي كالاتي :

١- قواعد ببليوغرافية (Bibliographic Databases) وهي اكثر انواع قواعد الاقراص انتشاراً واستخداماً ، وتحتوي على الوصف العام (الببليوغرافي) لمصادر المعلومات ، كجهة التأليف ، والعنوان الكامل ، والطبعة ، والناشر او اسم الدورية ، ومكان وسنة النشر ، والصفحة او الصفحات الواردة بها المعلومات ، ونبذة مختصرة عن المصدر (المستخلص) ورؤوس الموضوعات او الواصفات المهمة الدالة على موضوع او موضوعات المصدر ، وما شابه ذلك من البيانات الاساسية المطلوبة . ومن أمثلة هذه القواعد القاعدة الطبية (MEDLINE) وقاعدة مستخلصات الرسائل الجامعية المعروفة باسم (Dissertation Abstracts) وقاعدة المكتبات والمعلومات (LISA) .

٢- قواعد الادلة (Directory Databases) وتشتمل على معلومات مرجعية عن المؤسسات والمنظمات الرسمية وغير الرسمية ، المحلية والاقليمية والعالمية ، الثقافية منها والعلمية والتعليمية والتجارية ، مثل قاعدة دليل المؤسسات (Foundetion Directory) ودليل الدوريات (Ulrich,s International Periodicals Directcory) وما شابه ذلك .

٣- القواعد الرقمية والاحصائية (Numeric Databases) تشتمل اقراصها على معلومات احصائية سكانية او ثقافية او مالية .. الخ . ومن أمثلتها (Moody,s Corporate News) وقاعدة الاستشارات (INVESTEXT) . واستخداماتها محدودة في المكتبات ومراكز المعلومات .

العراقية والعربية ، خاصة في الاجابة على الاستفسارات المرجعية .

٤- قواعد النصوص الكاملة (Fulltext Database) وتشتمل أقراصها على المحتويات الكاملة للنصوص والوثائق والمطبوعات الأصلية مثل بعض دوائر المعارف (Academic , Gollier .. etc.) واخبار وكالات الانباء والصحف (AP News, Chicago Tribute) .

وهذه أيضاً محدودة (او معدومة) الاستخدام في المكتبات والمراكز المحلية والعربية في المنطقة ، لأن اكثر القواعد المستخدمة هي من نوع القواعد البيليوغرافية (انظر الملاحق والصفحات القادمة من البحث) .

بنوك وشبكات المعلومات المعتمدة على الأقراص

نعني بمصطلح بنك المعلومات (Information Bank) في دراستنا هذه المكتبات ومراكز المعلومات التي تقوم بتقديم خدمات معلومات متطورة ، عبر قواعد بياناتها البيليوغرافية وغير البيليوغرافية ، للباحثين والمستفيدين الاخرين ، من خلال عدد من قواعد البيانات المتوفرة على الاقراص المكتنزة (CD - ROM) او في ذاكرة حواسيبها والوسائط الاخرى التي تقتنيها . اضافة الى تأمين الوثائق الاصلية المطلوبة للباحثين ، بشكل مباشر من خزينها ومجاميعها ، او بشكل غير مباشر عن طريق استنساخها وطلبها من مكتبات ومراكز محلية او عربية او عالمية .

وعلى أساس ما تقدم فان مكتبات جامعة الموصل ، ومكتبة كلية الطب بجامعة صدام ، ومركز المعلومات والمعرفة العلمية في هيئة التصنيع في العراق، ومكتبة جامعة اليرموك في الاردن ، والتي ستمثل محور دراستنا في الصفحات القادمة ، تمثل كلها بنوكاً للمعلومات. الا ان قيام بنوك المعلومات هذه - وغيرها من بنوك ومراكز المعلومات - بتقديم خدمات أفضل للباحثين والمستفيدين يحتاج الى نظام تعاوني يقوم على الاستعانة ببعضها للبعض الآخر، مما يقود الى التحول الى نظام شبكة المعلومات .

وشبكة المعلومات (Information Network) تعني ربط اكثر من جهة - مكتبات أو مراكز معلومات - بوسائل اتصال حديثة ، سلكية قريبة المدى او لاسلكية بعيدة المدى عن طريق الخطوط الهاتفية والاقمار الصناعية ، سواء ما كان منها شبكات محلية (LAN / Local Area Network) أو شبكات اتصال عن بعد (Remote Communication Net work) . وباستخدام حواسيب مختلفة الاحجام والانواع والقدرة ، او مظاريف (Terminals) ذكية اوغير ذكية . مما يفيد الباحثين والمراكز والمكتبات في توفير كميات هائلة من المعلومات ، باقل الجهود واسرع وقت ممكن .

ثانياً : تجارب عالمية وعربية ومحلية في استخدام قواعد الأقراص .

تجارب اقليمية (عالمية)

تستطيع المكتبات ومراكز المعلومات التي تستخدم قواعد اقراص اقرأ ما في الذاكرة (CD-ROM) من التنسيق فيما بينها ، على طريق بناء نظام وطني او اقليمي يكون بشكل شبكة معلومات ، تعمل على تبادل المعلومات عن طريق قواعد الاقراص ، وخاصة القواعد البيليوغرافية منها. كذلك فانها تستطيع التعاون في مجال تأمين الوثائق الاصلية للمستفيدين . أما أهم التجارب الاقليمية العالمية ، والتي يمكننا الاستفادة منها على المستويين الوطني والعربي، التجارب الثلاثة الاتية [٩]:

١. تجربة استخدام الاقراص (CD - ROM) في مشروع الغذاء والتغذية IFNP / Food and Nutrition Pro-ject) التابع لجمعية الجامعات الافريقية (AAU/ Association of African Universities). فقد بدأت هذه

(9) Mark keybord . CD - ROM implementation in developing countries: Impacts and Pitfalls. IFLA Journal . 19,1993, 1. pp. 41-42 .

التجربة عام ١٩٩١ ، على طريق دعم أنشطة وخدمات المعلومات المتوفرة في الجامعات المشاركة في المشروع، بغرض تطوير قاعدة افريقية للغذاء والتغذية يكون مقرها في مدينة (اكرا)في غانا . وتقوم بتقديم خدمات المعلومات الى الجامعات المشاركة التي بلغ عددها (٩٤) جامعة . تعمل على التعاون في تبادل المعلومات . ويقوم بنك المعلومات ، او محطة الاقراص (CD-ROM) الذي أنشئ، في مدينة المقر، المار ذكرها بالتنسيق في هذا المجال . ويعتمد هذا النشاط التعاوني بالدرجة الاساس على قاعدة اقراص المستخلصات والتكنولوجياوعلم التغذية (FOOD SCIENCE AND TECHNOLOGY ABSTRACT/FSTA). وقد تم تدريب عدد من المتخصصين في مجال التوثيق واساليب تجميع ومعالجة البيانات واسترجاع المعلومات . ويقوم مركز المقر باصدار نشرة اخبارية (NEWSLETTER) عن الخدمات التي يقدمها الى الجامعات الاعضاء في المشروع وهنالك استمارة (قسيمة) طلب البحث توزع على الجهات المشاركة . ومقابل هذه الخدمة يطالب المركز الجامعات المشاركة تزويده بالوثائق والمواد التي لها علاقة بموضوعات الغذاء والتغذية التي تنتجها هذه الجامعات او تصدر عنها . وتكون هذه الوثائق والمواد الرصيد الجديد المضاف الى المدخلات والتحديثات لقاعدة الاقراص المذكورة والمنتجة محلياً .

٢- تجربة بنك المعلومات الزراعية المعروف باسم قاعدة الأقراص (Sesame) مثال اخر للتجارب الاقليمية في العالم والتي لها علاقة بالدول النامية . وهي قاعدة تخص الدول المتحدثة باللغة الفرنسية، في موضوعات الزراعة المختلفة . في مراكز التوثيق والمعلومات المتعاملة مع هذا الموضوع في كل من فرنسا ، وبلجيكا ، والكاميرون ، والجاد ، ومدغشقر ، والسنغال ، ودول الساحل الافريقي . ويجري التنسيق في هذه التجربة عن طريق مركز للتعاون الدولي لبحوث الزراعة

CIRAD / Centre de Cooperation Internationale en Recherche Agronomique le Developpement).

٣- تجربة بنك المعلومات الصحية المعروف باسم (LILACS CD - ROM) والذي يعمل على اقامة نظام شبكة تعاونية لدول امريكا اللاتينية والكاريبيان في معلومات العلوم الصحية (LATIN AMERICAN AND CARIBBEAN CENTER ON HEALTH SCIENCES INFORMATION / BIREME) ففي هذه التجربة تجمع كافة الوثائق والمعلومات المتعلقة بالعلوم الصحية في هذه المنطقة من العالم ، والتي تنتج في مختلف المعاهد والمؤسسات ، ويعاد تنظيمها لتتحول الى قاعدة بيانات بالاقراص المكتنزة (CD - ROM) .

ومن خلال هذه التجارب الثلاثة ، وغيرها من تجارب العالم الاقليمية في بناء بنوك وشبكات معلومات معتمدة على اقراص اقرأ ما في الذاكرة او الأنواع الاخرى من الاقراص المكتنزة ، وخاصة البيليوغرافية منها ، فانه تظهر حقيقة ازدياد طلب الباحثين والمستفيدين الاخرين على حصولهم على الوثائق الاصلية - او صور منها - من المكتبات ومراكز المعلومات المشاركة ، وذلك بضوء المعلومات البيليوغرافية المسترجعة من قواعد الاقراص المستخدمة . فانه من خلال الاشارة الى المصادر التي تخص موضوعاً معيناً ومن خلال استعراض مستخلصات هذه المصادر ، يتأكد الباحثون والمستفيدون الاخرون من حاجتهم الى المصادر والوثائق الاصلية التي يحتاجونها ، او نسخ مصورة عنها .

وعلى أساس ما تقدم فان قواعد وبنوك المعلومات المعتمدة على الاقراص المكتنزة (DC - ROM) يمكن ان تفيد في مجالات عدة أهمها : [١٠] .

١- الاستجابة الافضل لطلبات مجاميع المستفيدين ، حيث يمكن ان يصل الباحثون والمستفيدون الى اعداد كبيرة وشاملة من ملفات البيانات المخزونة باقراص اقرأ ما في الذاكرة (CO - ROM) .

٢- تطوير خدمات معلومات مصممة بشكل أكثر تخصصاً متناسبا مع حاجات المستفيدين ، معدة للاستخدام او الاسترجاع بأشكال متعددة ، كالاشكال المطبوعة ورقياً

(10) Ibid . p. 44

او على شكل اقراص ممغنطة او اقراص مرنة او أية وسائط نقل وحفظ المعلومات ، وبشكل يناسب الامكانات والتكاليف المالية .

٣. اجتناب أكبر عدد ممكن من المستفيدين . لأن هذا النوع من خدمات المعلومات المعتمدة على تكنولوجيا متطورة تستهوي الباحثين المستفيدين اكثر من الانواع والوسائط الاخرى التي توفرها المكتبات ومراكز المعلومات .

تجارب عربية / تجربة جامعة اليرموك (الاردن)

شرعت جامعة اليرموك بتقديم خدمات قواعد اقراص اقرأ (CD-ROM) للباحثين ، في عام ١٩٩٣ ، وذلك من خلال شبكة محلية (LAN) للمعلومات .حيث تقوم مكتبة الجامعة ، عن طريق [١١١] مشغل للأقراص (CD - ROM Drive) ، ترتبط بها أربع محطات تشغيل ، ثلاثة منها في المكتبة نفسها واخرى في بناية كلية العلوم بتقديم خدماتها المتطورة وعن طريق مجموعة من قواعد البيانات على الاقراص المتمثلة عناوينها بالاتي : [١١١] .

١- قاعدة الرياضيات MATHSCIENCE .

٢- قاعدة المستخلصات البيولوجية BIOLOGLCAL ABSTRACTS

٣- قاعدة البيليوغرافيا العالمية لجمعية اللغات الحديثة .

MODERN LANGUAGE ASSOCIATION INTERNATIONAL BIBLIOGRAPHY

٤- قاعدة مركز معلومات المصادر التعليمية

.EAIC / Educational Resources Information Centre

٥- قاعدة كشاف اسنادات العلوم الاجتماعية Social Science Citation Index

٦- قاعدة مستخلصات الرسائل الجامعية Dissertation Abstracts

٧- قاعدة الفيزياء Physics .

(١١١) مكتبة جامعة اليرموك . قسم قواعد البيانات . شبكة قواعد البيانات المنشورة على أقراص الليزر : دليل الباحث ، ١٩٩٣ . ص ١ - ٩

٨- قاعدة الالكترونيات والحاسب الالكتروني Electronics and Computing .

أما مجال استخدام شبكة تعاونية اوسع ، على مستوى الاردن والجامعات الاردنية او على المستوى العربي ربعض الجامعات العربية فهو غير موجود في الوقت الحاضر .

التجربة العراقية في استخدام قواعد الأقراص .

لقد خطت تجربة العراق في مجال استخدام قواعد الاقراص المكتنزة (CD.ROM) خطوات مهمة وجريئة في السنوات الاخيرة من عقد الثمانينات من هذا القرن . فقد كان العراق من اوائل الاقطار العربية التي اهتمت بهذا النوع من تقنيات المعلومات، قبل الحصار الاقتصادي والعلمي الذي بدت اثاره واضحة في السنوات الاخيرة التي تلت عام ١٩٩٠ . فقد كانت جامعة الموصل ومكتبتها المركزية سباقة في اقتناء عدد من الأقراص وتوفيرها للباحثين والمستفيدين الاخرين عبر وحدة المعلومات البحثية . ثم تعاقدت الجامعة المستنصرية على استحداث وحدة مماثلة ، وهكذا تعاقبت المؤسسات الاكاديمية والعلمية والبحثية في السير نحو استثمار هذا النوع من تقنيات المعلومات، على الرغم من الصعوبات الجمة التي احدثها ويحدثها الحصار .

ونستطيع تقييم جوانب اساسية من التجربة العراقية من خلال مجموعة من قواعد الاقراص المتوفرة في المكتبات ومراكز المعلومات ، التي تطورت خدماتها نحو استحداث شبكة تعاونية وطنية ، تضم مجموعة من الجامعات والمراكز ، ويلعب فيها مركز المعلومات والمعرفة العلمية دور نقطة الارتكاز (انظر الملحق رقم -١) واما اهم القواعد والاقراص المتوفرة فهي كالاتي : [١٢] .

أ - المكتبة المركزية لجامعة الموصل . وتتوفر فيها اكبر مجموعة لقواعد الأقراص المكتنزة أقرأ ما في الذاكرة ، حيث بلغت (٢٤) قاعدة اقراص هي في موضوعات

(12) CD-ROM Data bases available in Iraq on thè National Information Network . October , 1993 . . نشرة موزعة من قبل مركز المعلومات والمعرفة . .

شتى تعكس مجموع الاختصاصات في كليات ومعاهد الجامعة (انظر الملحق رقم -٢) *

ب - مركز المعلومات والمعرفة العلمية / هيئة التصنيع . وتتوفر في هذا المركز (٢٢) قاعدة أقراص في مواضيع علمية شتى . اضافة الى ذلك فان المركز - وكما بين سابقاً - يعمل كمنسق ونقطة ارتكاز (Focal Point) لشبكة الاقراص الوطنية التعاونية في العراق . ويبين الملحق رقم (٢) المرفق عناوين قواعد الاقراص الموجودة في هذا المركز.

جـ . كلية طب صدام . حيث تتوفر في مكتبتها ثمانية قواعد اقراص متخصصة ، وهي موضحة في الملحق المشار اليه أعلاه .

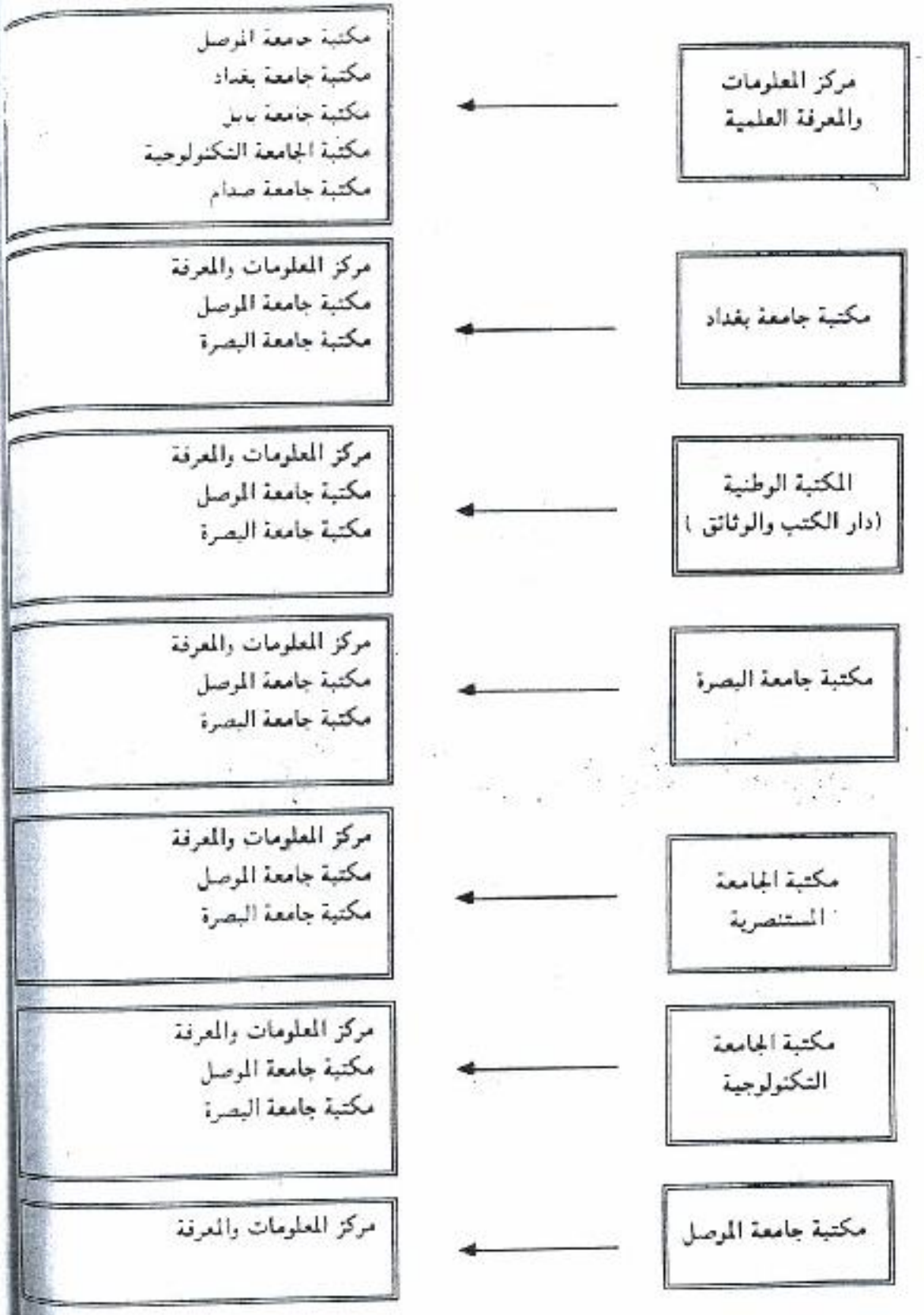
د- اضافة الى مجاميع متنوعة اخرى من قواعد الاقراص المتوفرة في مكتبات ومراكز اخرى مذكورة في الملحق (٢) .

وقد تبين للباحث انه بالاضافة الى التعاون في مجال تبادل المعلومات المتوفرة على الاقراص في مجال الشبكة الوطنية ، الا ان هنالك تركيز في التعاون وتبادل المعلومات يتكون من مجاميع متعددة مبينة في المخطط المرفق (٢) . حيث ان تبادل المعلومات يكون اساسه التسهيلات التي تحصل عليها المكتبات والمراكز من بعضها في اطار وسائل الاتصال الجيدة والموضوعات التي توفرها الاقراص .

فالمكتبات التي تمتلك رصيدا جيدا من الاقراص وقواعدها المختلفة ، مثل مكتبة جامعة الموصل ، تكون حاجتها للحصول على مزيد من المعلومات والتحري عنها محدودة بمركز المعلومات والمعرفة العلمية . اما مكتبة الجامعة المستنصرية فتحاول ان تستثمر امكانات الاثنين معاً ، مكتبة جامعة الموصل ومركز المعلومات والمعرفة ، وكما هو مبين في المخطط رقم (٢) المرفق .[١٣] .

* أظهر الملحق رقم (٢) أن مجموع قواعد الأقراص في مكتبة جامعة الموصل هو (٢٤) قاعدة . بينما أظهر مصدر اخر هو «دليل قواعد المعلومات في المكتبة المركزية لجامعة الموصل على أقراص الليزر الكنترة » والصادر في نفس الفترة . ان هنالك (٣٢) قاعدة أقراص . وعلى هذا الاساس فإن بعض القواعد هي غير متاحة عبر الشبكة الوطنية للمعلومات . وعددها (٨) أقراص .

(١٣) - ايمان فاضل السامرائي . التطبيقات الآلية في المكتبات ومراكز المعلومات في العراق : رسالة واقع . بغداد . الجامعة المستنصرية . ١٩٩٥. ص . ١٦



مخطط رقم (٢)

الاتصال والحاجة للمعلومات لبعض المكتبات والمراكز المشاركة بالشبكة العراقية لقواعد الأقراص المكـ

ثالثا :
الاستنتاجات والتوصيات

لقد خرج الباحث بعدد محدد من الاستنتاجات المهمة من تحليل المعلومات الواردة في متن الدراسة وملاحقها . وكذلك عدد اخر من المقترحات والتوصيات التي يراها مناسبة بضوء تلك الاستنتاجات ، وهي كالاتي :

أ - الاستنتاجات :

١- تمثل تقنية الأقراص المكتنزة ، والقواعد المخزونة عليها حالة متقدمة ومتطورة، ليس فقط قياساً بالوسائل التقليدية فحسب ، بل وحتى الوسائل الحديثة، كالبحث بالاتصال المباشر (Online Searching) والاقراص المغنطة (Magnetic Disks) وما شابه ذلك .

٢. ان قواعد البيانات المخزونة على اقراص اقرأ ما في الذاكرة (CD-ROM) من نوع القواعد البيليوغرافية (Bibliographic Databases) هي الأوسع انتشاراً واستخداماً في تجارب الدول النامية ، ومنها العراق والاردن ، على مستوى المكتبات ومراكز المعلومات .

٣- ان قواعد الاقراص البيليوغرافية تساعد كثيرا في التعرف على حاجة الباحثين الى الوثائق والمصادر الأصلية ، من خلال البيانات الاساسية عن تلك الوثائق والمصادر ، والمستخلصات الوافية عنها . وبذلك يستطيع الباحث والمستفيدون الاخرون من تحديد احتياجاتهم من الوثائق الاصلية أو صوراً مستنسخة عنها من المكتبات ومراكز المعلومات المتعاونة في اطار شبكة لمعلومات الاقراص .

٤- هنالك تعاون على المستوى الوطني في العراق بين المكتبات ومراكز المعلومات ، ضمن شبكة تعاونية لقواعد الأقراص المكتنزة اقرأ ما في الذاكرة . الا أن هذه الشبكة تواجه العديد من المشاكل والمعوقات التي يسببها الحصار الاقتصادي والعلمي على العراق، خاصة ما يتعلق بوسائل وتقنيات الاتصال وتحديث العديد من

قواعد الأقراص .

٥- ليس هنالك: تعاون ملموس على المستوى العربي - معروف لدى الباحث - على مستوى قواعد وشبكات الأقراص ، وعلى غرار ما هو موجود ومعروف في عدد من مناطق العالم ، وخاصة ما له علاقة بالدول النامية .

٦- لا تتوفر اية قواعد مصنعة او منتجة محلياً للنتاج الفكري والعلمي العربي ، سواء كان ذلك على مستوى الجامعات او المؤسسات العلمية والبحثية الاخرى ، بشكل أقراص مكتنزة ، وعلى غرار ما هو منجز في عدد من دول العالم ، ومنها بعض الدول النامية .

ب- التوصيات والمقترحات

١- يقترح الباحث تنظيم تعاون بين المكتبات ومراكز المعلومات ، عن طريق بناء شبكة معلومات في عدد من الاقطار العربية ، وكبداية بين العراق والاردن ، وان يكون هذا التعاون على مراحل ، يمكن ان تكون كالاتي :

المرحلة الأولى : تتبين التعاون على المستوى الوطني (القطري) بين بنوك معلومات مكتبة جامعة الموصل ومركز المعلومات والمعرفة العلمية والمكتبات والمراكز العراقية الأخرى ، ومحاولة معالجة المشاكل والمعوقات التي تواجهها الشبكة بالامكانات المحلية - أثناء الحصار وبعد انتهاءه - أو بامكانات عربية وعالمية اخرى متاحة .

المرحلة الثانية : تعاون وتنسيق مبرمج ومنظم بين التجربة العراقية في بناء بنوك وشبكة معلومات على أقراص اقرأ ما في الذاكرة من جهة ، وبين تجربة جامعة اليرموك في الاردن .

المرحلة الثالثة : العمل على انضمام كل من الجمعية الملكية الاردنية وجامعة العلوم والتكنولوجيا والجامعة الاردنية وجامعة العلوم التطبيقية (واية جامعات اردنية مهيئة اخرى) الى شكبة اقراص وطنية تعاونية . وبعبارة اخرى توسيع الشبكة المحلية (LAN) الموجودة في مكتبة جامعة اليرموك الى شبكة أوسع مداراً .

واستثمار امكاناتها والامكانات المتاحة ، او التي يمكن ان تتاح ، في المكتبات والمراكز الاردنية الأخرى .

المرحلة الرابعة : دراسة امكانات التوسع على المستوى العربي مع أقطار عربية اخرى مثل سوريا ولبنان ومصر وليبيا واليمن .. الخ .

٢- العمل ، بكل الاتجاهات والامكانات المتاحة ، على دعم مشاريع التعاون والتنسيق في مجال استثمار قواعد الاقراص المكتنزة ، من قبل المنظمات والاتحادات العربية الموجودة في المنطقة مثل اتحاد الجامعات العربية واتحاد مجالس البحث العلمي العربية ، اضافة الى مركز التوثيق والمعلومات بجامعة الدول العربية ، وادارة التوثيق والمعلومات بالمنظمة العربية للتربية والثقافة والعلوم .

٣-وضع الامكانات المتاحة ، فنياً وبشرياً ، لكل من جمعية المكتبات الاردنية والجمعية العراقية للمكتبات والمعلومات ، في خدمة مثل هذه المشاريع التعاونية الوطنية منها والقومية ، وبطرق متعددة ، مثل التوسع في دورات التدريب والتأهيل والاعلام والتوعية باتجاه اهمية التعاون ، وعقد ندوات وحلقات دراسية مخصصة لمثل هذه الموضوعات والمشاريع التعاونية .

قائمة المصادر

١- جامعة الموصل ، المكتبة المركزية / دليل قواعد المعلومات في المكتبة المركزية لجامعة الموصل على أقراص الليزر المكتنزة . الموصل ، د.ت، ٣٦ ص .

٢. السامرائي ، ايمان فاضل ، الأوعية المتعددة وتطور الأقراص منذ عام ١٨٧٧ وحتى عام ١٩٩٢ . المجلة العربية للمعلومات، مج ١٥ ، ١٩٩٤ ، ص ٨٢ - ١٢٣ .

٣- السامرائي ، ايمان فاضل ، التطبيقات الالية في المكتبات ومراكز المعلومات في العراق .رسالة دكتوراه مقدمة الى قسم المكتبات والمعلومات / الجامعة المستنصرية . ١٩٩٥ ، ٢٥٠ ص .

٤. الشربجي ، نجيب . الأقراص المتراصة : تفاعلات . مجلة رسالة المكاتبة . مج ٢٨ ، ع١ و ٢ كانون أول ١٩٩٣ ، ص .

٥- قنديلجي ، عامر ابراهيم . استخدام الحواسيب في خدمات المعلومات : ملامح عن التجربة العراقية - المجلة العربية للمعلومات . مج ١٥ ، ع١ ، ١٩٩٤ ، ص ٥٥ - ٨١ .

٦- قنديلجي ، عامر ابراهيم . بناء شبكة وطنية للمكتبات ومراكز المعلومات في العراق : المشاكل والحلول المقترحة . ندوة نظم المعلومات في القرن الحادي والعشرين ، جامعة اليرموك ٢٨-٣٠/ ١٩٩٣/٦ ، ٢٢ ص .

٧- مكتبة جامعة اليرموك . قسم قواعد البيانات . شبكة قواعد البيانات المنشورة على أقراص الليزر : دليل الباحث . ١٩٩٣ ، ٩ ص .

8. Akeroyd. John. CD-ROM networks . The Electronic Library - vol. 9 . no.1. Feb. 1991 . pp. 21-25

9. CD-ROM Databases available in Iraq on the National Information Network . Oct. 1993 . 8 p (نشرة موزعة من قبل مركز المعلومات والعرفة : هيئة التصنيع)

10. Silver Platter. 1993 Directory of CD-ROM Databases . Northwood , MA . Silver Platter Information . 1993 . 31p.

1- PAAM 97

The 2nd International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi - Agent Technology .
21 - 23 / 4 /1997, Lancashire U.K.

2 - Symposium 1997 on Safety - Critical Systems
4 - 6 / 2 / 1997 Brighton U.K.

3 - The 5th International Conference in Central Europe on Computer Graphics and Visualization 97 .
10 - 14 / 2 / 1997 ، plzen ، Czech . Republic .

4 - 1st Euromicro Working Conference on Software Maintenance and Reengineering .
17 - 19 / 3 / 1997 - Berlin - Germany .

5 - CHI - 97
Conference on Human Factors in Computing Systems
22 - 27 / 3 /1997
ACM . Atlanta U. S. A.

6 - ICCC / IFIP Conference on Electronic Publishing
14 - 16 / 4 / 1997 IEEE , Kent , U.K.

7 - IFIP / SEC 97 - 13th International Information Security Conference
15 - 16 / 5 / 1997 Copenhagen - Denemark .

8 - ICSE 97- International Conference on Software Engineering .
18 - 23 / 5 / 1997 IEEE , Boston , U . S . A

92. odress,instructionodresse dinstrution		عنوان التعليمة موقع التخزين المحتوي على تعليمة البرنامج
93. odress,machine	odresse mochine	عنوان الي عنوان مطلق بدون دليل ، يمكن للحاسب التعامل معه مباشرة
94. odress modifiction	modifiction dodresse	تعديل العنوان تغيير عنوان موقع التخزين بالذاكرة باستعمال طريقة معينة يحددها البرنامج
95. oddrss, mulli	oddress differee	العنوان المتعدد يحتوي على اكثر من عنوان
96. odress, multipple computer	colculmeur is shesses multiples	متعدد العنوان
97. oddress . one or single	odrsse unique	عنوان مفرد نظام يشتمل على عنوان واحد وعملية واحدة في جميع التعليمات
98 . oddress. one plus one	odresse un plus un	عنوان الواحد مع واحد العنوان المتضمن شطرين اولهما يمثل موقع تخزين البيانات والاخر عنوان التعليمة
99 . oddress part	lo portie odresse	قسم العنوان جزء التعليمة الذي يشتمل على العنوان
100 . oddress. reference	odress de reference	عنوان المرجع وهو عدد يمثل مكان العنوان

العنوان العائم العنوان المصوغ بشكل قابل لتغيير ليصبح العنوان الفعلي	82. adress,floating	adresse flootite
بنية العنوان تأليف العنوان ضمن تعليمات البرنامج	83. adress format	we format de dadresse
توليد العنوان يستعمل مع عناوين على اجهزة الوصول المباشر ويمثل عملية احتساب القيود بدلالة مفاتيحها	84.adress generation	geneution dadresse
عنوان فوري وهو دليل مباشر الى العنوان الفعلي في الذاكرة لتعليمـــة مـــا	85. adress,immediate	adresse immediate
عنوان مفهرس عنوان يتغير بتغير قيم أحد مسجلات الحاسب	86. adress,indexed	adresse indexede
عنوان غير مباشر عنوان وسيط يؤدي الى عنوان اخر (العنوان الفعلي عادة)	87.adress,indirect	adresse indirecte
المعنونة وضع العنوانات	88.adressing	adressinge
عنونة ثابتة الموضع	89. adressing,fixed – position	adressinge dune position fixe
عنونة نسبية	90. adressing,relative	adressugerelatil
نظام المعنونة الاسلوب المتبع في تحديد عناوين مواقع التخزين بالذاكرة	91. adressing,system	systeme dadressuge

69. add time	temps diaddition	زمن الجمع الوقت اللازم لاجراء عملية جمع واحدة
70, address	adrese	عنوان (موقع التخزين) رقم أو اسم حرفي يدل على موقع التخزين في ذاكرة الحاسب (التخزين الداخلي)
71, address,absolute	adresse absolue	عنوان مطلق (انظر المصطلح رقم ٥)
72. address,arithmetic	adresse arithmetique	عنوان حسابي العنوان الذي يتم الحصول عليه بلجراء احدى العمليات الحسابية بعنوان اخر
73, address,base	adresse de base as base	العنوان الاساس عنوان يستعمل بداية عنوانات لاحقة
74, address,constant	adresse de base	العنوان الثابت (انظر المصطلح رقم ٧٣)
75, address control	commande diadresse	ضبط العنوان
76, adress development	calcul dadresse	تطوير العنوان
77, address,direct	adresse dadresse	عنوان مباشر
78, adress,memory	memoire adressble	ذاكرة معنونة
79 address,effective	adresse effective	العنوان الفعال عنوان تم تحيلة بطريقة ما ليشير الى موقع تخزين داخلي (العنوان الفعلي)
80. adress,external device	dispocilif externe dadressuge	عنوان موقع تخزين خارجي
81. adress,first level	adresse premier niveau	عنوان مباشر (من المستوى الاول)

58. addend	NOMBRE a ajouter	العدد المضاف
59.adder	additionneur	الجامع
		وهو جهاز يجمع الاعداد المدخلة الى
		الحاسب
60. adder circuit	circuit dadditionneur	دائرة الجامع
		توجد في وحدة الحساب ويمكنها
		اداء الجمع بالترقيم الثنائي
61. adder half	demi additionneur	جامع نصفي
		دائرة تجميع رقمين
		ثنائيين لتعطي رقمين ثنائيين
		احدهما المجموع وثانيهما
		المرحل
62. adder full	additionner entier	جامع تام
		دائرة تجمع ثلاثة ارقام ثنائية
		اثنتان منها يمثلان رقمين
		والثالث يمثل المرحل من الموضع
		السابق
63. adder substracter	additionneur – soustracteur	جامع طارح
64. adding machine	mouchine is additionner	الة الجمع
65. addition	addition	الاضافة (الجمع)
66. addition table	table – daddition	جدول الجمع
		منطقة في ذاكرة الحاسب تحوي
		جدول اعداد تستعمل في
		التفتيش
		عن الجمع
67. additional characters	caracteres additives	رموز اضافية
		علامات خاصة زيادة على
		الحروف والارقام ورموز
		السيطرة
68. add – listing machine	additionneuse	الة جامعة طابعة

38. acoustic delay line	ligne arctard acoustique	عائق صوتي
39. acoustic memory	memoire acoustique	ذاكرة صوتية
40. acoustics	iaccoustique	علم الصوت
41. acoustic treatment	insonorisation	عزل صوتي
42. acoustic store	memoire acoustique	وحدة تخزين صوتية
43. acronym	dureviation	تسمية مختصرة
44. activation	declenchement;commiande	تنشيط
45. active element	element actif	عنصر نشط
46. activity	activite	نشاط
47. activity level	niveau dactivite	مستوى النشاط
48. activity ratio	rapport dactivite	نسبة النشاط
49. actual address	adresse effective	عنوان فعلي
50. actual code	code effective	الجفرة (الشفرة) للمطلقة
51. actual decimal point	point decimal effectif	علامة عشرية فعلية
52. actual key	cle reelle	مفتاح فعلي
53. adapter	adaptateur	مكيف
54. adapter	adaptateur de ligne de	مكيف خط الاتصال
55. adapter . transmission	adaptateur de transmission	مكيف البث
		يتولى ربط الاجهزة البعيدة والمحلية بمكيف البيانات وكذلك الضوابط الضرورية وذلك لنقل البيانات الى او من وحدة المعالجة
56. adapters . dataphone	adaptateurs de telecommunication	مكيفات الإتصال الهاتفي
57. adaptive control system	systeme du controle diadtation	نظام الضبط المتكيف نظام الضبط ترصد توصد فيه المؤثرات ويتكيف ذاتيا ليلائم المتغيرات المحيطة به وذلك بتغير المعالم الداخلة

22. accession number	numero de referene	رقم المرجع
23. acces methods	methodes d,acces	طرق الوصول يقصد بها أساليب الوصول الى مواقع التخزين المراد التعامل معها (متتابعة او مباشرة او فورية)
24. access mode	mode d,acces	طراز الوصول
25. access , simultaneous	access simultane	وصول متزان
26. access, random	acces selectif	عشوائي
27. access,sequential	acces sequentiel	وصول تتابعي الحصول على البيانات من جهاز ادخال / اخراج بطريقة تسلسلية
28. access,serial	acces en serie	وصول تسلسلي
29. access,time	temps d,acces	زمن الوصول الزمن اللازم للوصول الى مواقع التخزين
30. accounting machine	tabulatrice	آلة محاسبة الآلة التي تقوم بالعمليات المحاسبية المختلفة
31. accumulating counter	compuler tabaliseur	عداد التجميع
32. accumulative card	carte cumule	بطاقة جامعة
33. accumulator	accumulateur	مجمع مسجل خاص وجزء من الوحدة الحسابية للحاسب وتوضع فيه نتائج العمليات الحسابية والمنطقية
34. accumulator register	registre de cumul	مسجل مجمع
35. accuracy	e.xactitude	دقة درجة التقارب بين النتيجة الحاصلة والقيمة الحقيقية
36. accuracy-control	systeme de controle dexactitude	نظام ضبط الدقة
37. A-C dump	decharge de courant alternatif	تفريغ التيار المتناوب

مصطلحات معربة

14. accept stacker	case de r.eception normale	بيت القبول وهو المكان تكدس فيه المستندات المقبولة بعد ان تمر بمحطة القراءة لي قارئ الحاسب
15. access	acces	وصول الوصول الى مواقع التخزين بغرض التعامل معها
16. access arm	bras d,acces	ذراع الوصول الذراع المستخدمة في وحدة الاسطوانات المغنطة والتي تحمل رؤوس القراءة والكتابة على الاسطوانات
17. access control registers	registre de controle d.aces	سجلات ضبط الوصول
18. access control	mots controle daccs	كلمات ضبط الوصول
19. access cycle	cycle dacces	دورة الوصول
20. access;direct	acces direct;acces selectif	وصول مباشر وصول الى البيانات المسجلة على وحدات التخزين الممغنطة التي لاتتطلب تتابعا في تسجيل البيانات وانما تتطلب عددا ثابتا من العمليات (لازمنا ثابتابالضرورة) بغض النظر عن الموقع المراد الوصول الية
21. access. immediate	acces direct	وصول فوري وصول الى مواقع التخزين يستغرق زمنا واحدا .

مصطلحات معربة

1. abacus	abaque.boulier computer	معدات الة بسيطة تستعمل لاجراء العمليات الحسابية الرئيسية، وتتكون من عشرة اسلاك تتحرك
2. abnormal condition	anomalie	حالة شاذة
3. abort	avorter.faire avorter arreter une exploition avant terme	اخلاء اخلاء برنامج قبل نهايته المعلومة
4. above-plateur device	dispositif dalimentation popier	مغذي الورق
5. absolute address	adresse absolue	عنوان مطلق
6. absolute code	code absolu	جفرة (شفرة) مطلقة وهي نظام من الرموز يستعمل للدلالة على العناوين المطلقة والعمليات الاساسية في نظم الحاسبات
7. absolute coding	codage absolu	جفر (تنظم شفري) مطلق
8. absolute error	erreur absolue	خطأ مطلق
9. absolute instruction	instruction absolue	تعليمة مطلقة
10. absolute value computer	ordinateur;caldinateur; pour valeur absolue	حاسبات القيمة المطلقة حاسب يعالج المعطيات المعبرعنها في قيمها الكاملة
11. absract	condense;resume	خلاصة
12. abstractor	resumeur	ملخص
13. acceleration	temps dacceleration	زمن التعجيل

Acknowledge

I would like to express my appreciation to Dr. Alaa H. Alhamami, the Head Department of the Computer Science at Al-Rafidain University Colledge for his valuable suggestions and advice provided during this study.

Thanks are also due to Dr. Hilal M. Yousif, the dean of Al-Rafidain University Colledge for his help and support.

Finally, a note of thanks to the members of the computer center / Al-Rafidain University Colledge for their help during typing this study.

REFERENCES

(1) Aho,A. L., Hopcroft, J. E and Ullman, J. D. Data Structures and Algorithms. Addison Wesley, 1983

(2) Bayer, R and E. McCreight. "Organization and Maintenance of Large Ordered Indexes." ACTA Information, 1 (1979), 173-189.

(3) Dolan, K. A. and Kroenke, D. Database Processing Fundamental. Design Implementation. Science Research Associates, Inc. 1988

(4) Knuth, Donald E. The Art of Computer Programming. Vol. 3. Addsion- Wesley, 1973

(5) Kruse, R. L. Data Structure and Program Design. Prentice-Hall, 1984

(6) Sahni, S . and Horowitz, E. Fundamentals of Data Structurs. Computer Scimce Press, Inc. 1987

(7) Nelmat, M. K. Search Mechanisms for Large Files. UMI Research Press, 1981

(8) Salton, G. and McGill, M. Introduction to Modern Information Retrieval. McGraw-Hill. 1983

(9) Standish, T. A. Data Structure Techniques. Addison-Wesley, 1980

(10) Van Doren, J. R. "Information Organization and Retrieval." (Unpub. Class notes, Oklahoma State University. 1983)

(11) Wiederhold, G. database Design. McGraw-Hill. 1983

4. Discussion

The advantage of this method can be found through the statistical information about the tree utilization, also the number of node accesses during overflows and splits. If the tree is stored on a secondary storage device such as disc, each of these node accesses represent a disc access which can be a very important factor to show the overall efficiency of the B-tree.

Tree utilization and node accesses are very important in the construction and maintenance of B-tree but this is not the only area in which these play a vital role. When searching for a key in a B-tree the maximum length of search path is the number of levels in the tree and of course this means a node access at each level. Therefore the predication of the number of levels that will result in the tree can be very important factor in choosing a node size for the tree.

```
    else
       move father key into right brother
       move largest key from left brother to
                                father key
    end
    replace nodes into the tree
  else
    move father key into leaf node
    replace father node into the tree
    move keys from brother into leaf node
    replace leaf node into the tree
    remove brother node from the tree
    if father node is not root
       call SIZE_CHK
    else
       if root node is empty
          root = leaf
          decrement levels in tree
          remove father node from tree
       end
     end
  end
end SIZE_CHK


OUTPUT
  print # keys in tree, order of tree, # of levels
/* calculate tree utilization */
  c_nodes = 100 *(nodes_used/max. # of nodes)
  c_usage = 100 * (actual_# of keys in tree /
 (nodes_used * max._# ofkeys_allowed_in_one_node)
  print c_node, c_usage

/* calculate leaf node utilization */
  print # of keys contained in leaf node
  print total_utilization_of_leaf_node =
     (keys_in_leaves / (# of leaves * max._# keys
       allowed in one node) * 100
  print split_overflow = (# of overflow /
      (actual_# keys in tree)) * 100
  print  # of overflow, keycnt,
       # of sib_rel_overflow,
       # of sib_rel_split
end OUTPUT
```

```
HEAD_NODE
   increment number of levels in tree by 1
   generate new node for the tree
   insert the key into the new node
   set up the two pointers from this node
   make this node as the root of the tree
   place the node into the tree
   return
end HEAD__NODE


DELETE
   call SEARCH
   if key not in the tree
      return
   end
   if key in a leaf node
      remove key from leaf node
      replace node into the tree
      if node not root
         call SIZE_CHK
      else
         if tree empty
            root = level = 0
         end
      end
   else
      access leaf with next largest key
      replace key being deleted with next largest key
      replace non_leaf node into the tree
      reduce size of leaf node
      replace leaf node into the tree
      call SIZE_CHK
   end
end DELETE


SIZE_CHK
   if node not too small
      return
   end
   if node is root
      return
   end
   access father node
   if leaf node right-most sibling
      access left brother
   else
      access right brother
   end
   if catenation possible
      if left brother too small
         move father key into left brother
         move smallest key from right brother to  father key
```

```
                    else
                        if left brother not exist
                            call 3_W_SPLIT
                        end
                    end
                end
            else
                perform overflow to the right
                replace nodes into the tree
                return to INSERT
            end
        end
    end
  end
  end do
  end
end OVERFLOW


3_W_SPLIT
  if not 3_w_split specified
    replace nodes into the tree
  else
    set up nodes for 3_way split
    generate a new node
    calculate size of the 3 nodes
    move key from left node into new center node
        leaving left node in final form
    move father key into new center node
    move key from left node into father key
    complete the updating of the new center node
    shift father node to make room for new key
    if 3-2 tree  /* # of nodes <= 3 */
      take father key from center node
    else
      take new father key from right node
      complete the updating of the right node
    end
    replace three brother nodes into the tree
    if father size not exceeded
      replace father node into the tree
      return to INSERT
    else
      return to SIZE_EX
  end
end 3_W_SPLIT
```

```
        if node size exceeded
            call SIZE_EX
        else
            replace current_node in tree
            return to B_TREE
        end
      end do
    end
  end
end INSERT


SIZE_EX
  do forever
    if overflow insertion
      determine overflow direction
      get father node
      if not underflow
        call OVR
      end
    end
    /* two way split */

  end do
end SIZE_EX


OVERFLOW
  determine position of father key
  if (overflow direction is left and left brother exist) OR
       (overflow direction right and right brother exist)
    do forever
    /* Left */
      access brother node
      if brother not full
        perform overflow to the left
        replace nodes into the tree
        return to INSERT
      else
        /* brother full */
        if not both direction
          call 3_W_SPLIT
        else
          if right brother not exist
            call 3_W_SPLIT
          else
            /* Right */
            access brother node
            if brother node exist
              if not both direction
                call 3_W_SPLIT
              else
                if other direction
                  call 3_W_SPLIT
```

```
B_TREE
  do forever
    read transaction
    if transaction is insert
        call INSERT
    if transaction is search
        call SEARCH
    if transaction is delete
        call DELETE
    if transaction is output
        call OUTPUT
    if transaction is end
        return to main
  end do
end B_TREE

SEARCH
  next_node = root
  do forever
    access next_node
    perform binary search on node for key
    if key found
       save position where key is found
       return
    else
       if leaf_node
          save position where key should be
          return
       else
          determine next_node
       end
    end
  end do
end SEARCH


INSERT
  if tree empty
     current_node = 0
     key_current = key _current + 1
     call HEAD_NODE
     return to B_TREE
  else
     call SEARCH
     if key found
        return to B_TREE
     else
        do forever
           insert new key into current_node
           if first_pass
              key_current = key_current + 1
              turn off first_pass switch
           end
```

Before inserting the key, the node size must be checked and if the node size has been exceeded, then either a two way split or an overflow will be performed according to the status of the input parameters provided in the main routine.

When the deletion of a key from the tree is requested, the procedure DELETE is invoked. First the search procedure is called to position the key in the tree. If the key is not found then the delete request is ignored. Otherwise one of two methods is used to delete the key: one method for leaf nodes and another for nonleaf nodes.

OUTPUT procedure print the following information:

- The number of keys in the tree.
- The order of the tree.
- The number of levels in the tree.
- The number and percentage of the available nodes used in the tree.
- The total utilization of the tree.
- The frequency count of the number of keys in the leaf nodes.
- The frequency utilization of the leaf node in the tree.
- The number and percentage of overflows that occured during all insertions in the tree.
- The total number of sibling nodes referenced during all of these overflows.
- The total number of sibling nodes referenced during all splits that have occured in the construction of the tree.

3.1 Algorithms

```
main
    call read-parameter
    if #-of-trees <=0
        #-of-trees = 1
        print parameter
        MAX = 2 * MNM
        ORDER = MAX + 1
        NN = ⌈[(3 * NUM) / (2 * MAX) * 1.1)⌉
        for tree-cnt = 1 to #-of-trees
            keycnt = nodes_used=prev_output_size= 0
            #_of_overflows = #_sib_ref = 0
            #_sib_ref_ovfw= #_sib_ref_split= 0
            root = level = 0
            call B-TREE
end main
```

3. Proposed Algorithm

This proposed algorithm is dedicated to the problems revolving around the structure of a B-tree, the methods of insertion and deletion in B-trees. For instance, methods of handling overfull nodes. If a node is overflowing and it is the leftmost or rightmost son of a father then an overflow may take place in only one direction. A possible variation is to attempt an overflow to a cosin node before performing a node split if the brother node is full. Also if a node and its two brothers are full, a four way split requires dividing the contents of the three full nodes and producing four nodes, each of which are approximately three-fourths full. An important condition on B-tree that must be taken into consideration has been stated by Kruse (5), "The condition that all leaves be on the same level forces a characteristic behavior of B-tree. B-trees are not allowed to grow at their leaves; instead they are forced to grow at the root."

The purpose of this algorithm is to perform the three basic operations possible on B-trees (search, insert, and delete) and provide an analysis of tree utilization by using different method of insertion with trees of different order.

The algorithm is structured so that the procedures that perform the tree operations are combined in a basic package (main program). All auxiliary procedures are external to this package so that they may be conveniently modified.

The main algorithm (MAIN) performs two functions: read in the parameters needed and then call the B_TREE procedure that perform various operations on the tree.

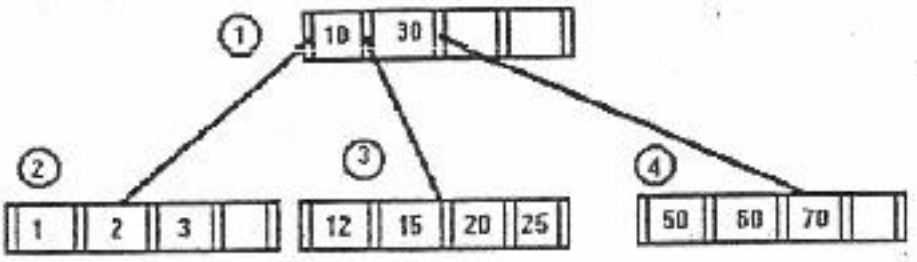
The main parameters needed are:

- The minimum number of keys allowed in one node (MNM).
- The maximum number of keys to be in the tree (NUM).
- A switch which indicates if the method of overflow is to be used during insertion (OVERFLOW).
- The overflow direction (left or right overflow).
- A switch which indicate whether a two way split or three way split is to be used (3_W_SPLIT).
- The number of trees to be generated (#_of_trees).

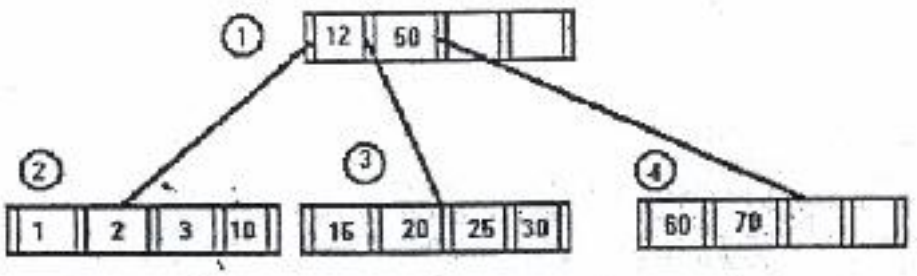
The B_TREE procedure reads a transaction code to determine one of the four possible operation (insert, search, delete, and output) is to be performed and call the correct procedure to carry out this task.

SEARCH is the routine that searchs the tree for a particular key. The search originates with the root node. A binary search is used when "looking" for the key.

INSERT is the procedure that is called to insert a key into the B_tree. Each time a new key is added to the tree, the key count is increased by one. The routine HEAD_NODE is called only when attempting to insert a key into an empty tree.



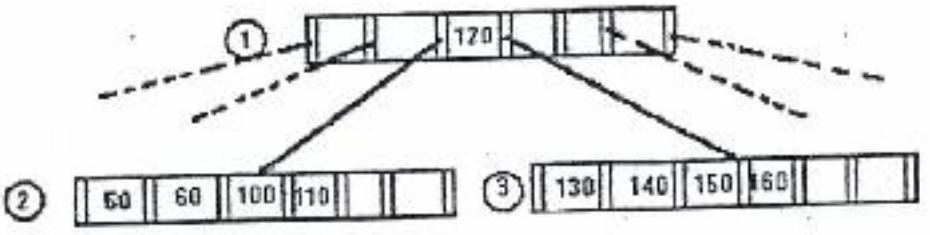
b) After Overflow to the right



c) After Overflow to the Left

Figure 11. An Order Five B-Tree Illustrating the Overflow Insertion Technique

It is important to note at this point that an overflow (like an underflow) changes only the contents and not the size of the father node. For this reason, overflows will not propaget back up through the tree.



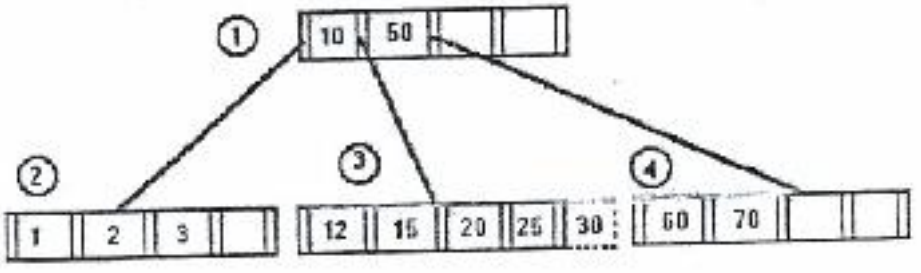
b) After Underflow

Figure 10. A Subtree of an Order Seven B-Tree Illustrating the Underflow Process During Deletion

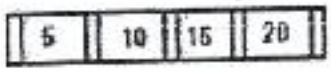
2.4 Insertion With Overflow

After looking at the basic insertion algorithm and deltion with underflow, a method of insertion using an overflow technique to handle nodes when they become too full can be developed. This method comprises movement of keys between brothers. Overflow is very similar to underflow. When a node becomes too full because of an insertion of a key, an attempt is made to redistribute the keys of the overfull and a brother before resorting to a node split. Overfull, like underflow, can be considered to be a stepwise process involving the father key aa well as the keys in the brothers.

Figure 11a is an example of a tree after a new key has been inserted but before execution of an overflow. Figure 11b illustrates the tree after completing an overflow to the right. Overflow can go in either direction to either brother; therefore an overflow to the left can be performed on the tree in Figure 11a resulting the tree in Figure 11c.



a) Before Overflow

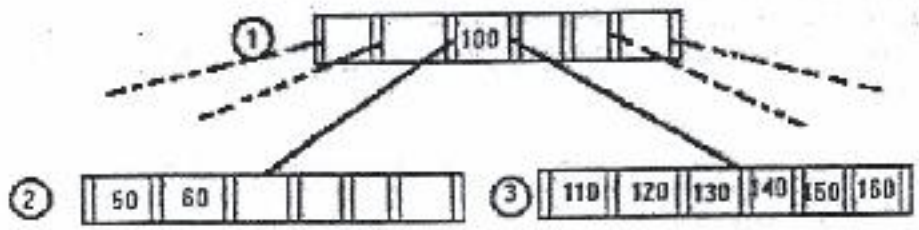


b) After Deletion of Key 7

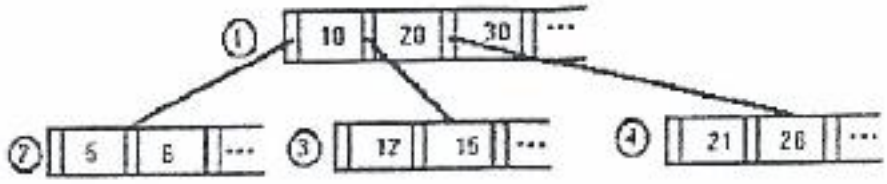
Figure 9. The deletion of a Key From an Order Five B-tree Which Results in the Construction of a New Root Node.

If catenation is precluded by the upper limit on the node size, an "underflow" must be performed on the nodes involved.

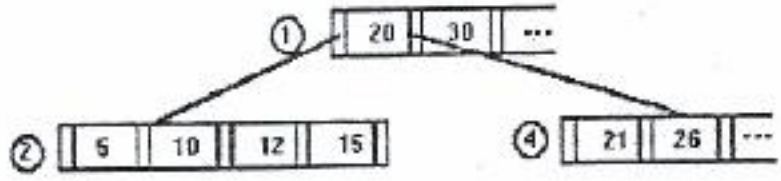
The underflow process comprises equal distribution of the keys between the two nodes. The father key as well as the keys of the two brothers are involved in this distribution. The underflow algorithm can be described as a stepwise process where the father key is moved to the node that is too small and a key from its brother is used to replace the father key. This is repeated until the two brothers are the same size (or as near as possible). For example, let Figure 10a represents the configuration of part of an order 7 B-tree after the deletion of a key. The "underfull" node in this case is node 2 and it is the left brother. Figure 10b represents the same portion of the tree after the underflow has taken place.



a) Before Underflow



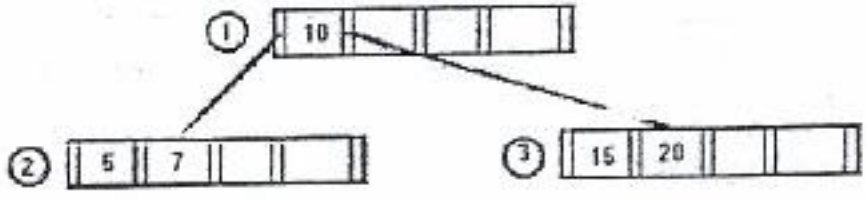
a) Before deletelon of Key 6



b) After deletelon of Key 6

Figure 8. A Subtree of An Order Five B-Tree Illustrating the Catenation Process Used During Deletion

Since the father node has been reduced in size it must be tested to see if it has fallen below the minimum node size. If so, either catenation or underflow must be performed on this node. catenation, as in the case of two splits during insertion, may be propageted through the tree toward the root node. If, during a catenation, the father of the two brothers being combined is the root of the tree and it contains only one key, then the node formed by the catenation process is the new root of the tree and the number of levels in the tree is reduced by one (Figure 9).



a) Before Deletion of Key 7

2.3 Deletion

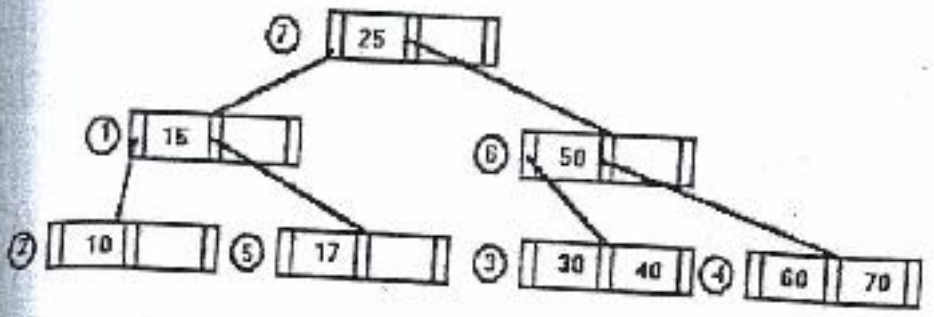
Just as in the case of insertion, the deletion of a key from a B-tree is straightforward unless one of the basic properties of B-tree is violated. Because of the characteristic of B-tree being constantly in balance, a key may have to be removed from a node which may cause the size of this node to fall below the minmum node size.

There are two types of nodes in a B-tree from which a key may be deleted: a leaf node or a non-leaf node. Deleting from a leaf node does not cause immediate problems because the deletion may take place without regard to loosing a pointer since all pointers in a leaf are null.

A problem does arise however, when attempting to delete a key from a non-leaf node or more directly a node with non-null pointers. This implies that a node or possibly a complete subtree below this node will be lost. To avoid this, the key that is being deleted is merely replaced by the next largest (or next smallest) key in the tree. The next largest key is the smallest key in the subtree pointed to by the pointer immediately to the right of the key being deleted. The smallest key in a subtree is found by the following pointer zero down through the tree until a null pointer is encountered, indicating a leaf. The first key in this leaf node is the smallest key of the subtree. (Similarly, the next smallest key in the tree is the largest key in the leaf subtree). The only problem left to handle in the deletion process is if the node size of the leaf that is reduced falls below the minimum node size allowed in the tree. If this condition does not occur then the deleteion of the key is completed. However, if this problem arises, one of two possible corrective actions must be taken. These two actions, which are mutually exclusive, are known as:

- catenation, and
- underflow

In the case of catenation, the node that is too small and a brother node (a node which has the same father and is immediately adjacent to the node in question) are catenated or joined together to form a single node. This process can only taken place if the sum of the numbers of keys in the two brother nodes being "catenated" is strictly less than the maximum number of keys allowed in any node in the tree. The catenation will decrease the number of nodes in the tree by one so the pointer to this lost node also must be removed. The father key is the key in the father node that logically falls between the two brothers being combined. Therefore the father key also becomes part of the new node being formed which means that it is deleted from the father node along with the no longer needed pointer. This is illustrated by the subtree shown in Figure 8. When key 6 is deleted from node 2 in Figure 8.a, node 2 and 3 along with father key 10 will be combined to form a new node 2 as shown in Figure 8b.



b) After Insertion of Key 17

Figure 7. An Order Three B-Tree Illustrating the Addition od a New Root Node Which Adds to the Number of Levels in the Tree

Careful examination of the two way split process, illustrates that the node involved in the split will always remain on the same level (relative to the bottom). For this reason, once a node is calssified as a leaf node, then it will always be a leaf as long as it is part of the tree.

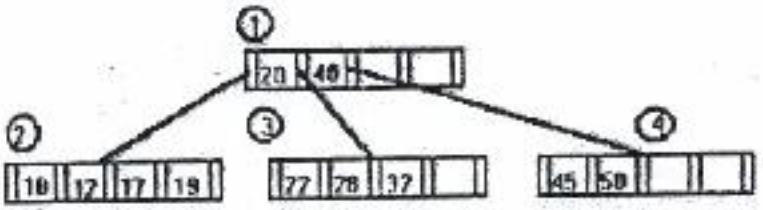
As previously stated, this process of two way splitting may propagate back through the tree to the root node. If at any point of this propagation of splits a father node does not exceed the maximum node size after the insertion, then the process stop immediately. It is not necessary that splitting be propagated all the way to the root node of the tree.

Empirical evidence given by Bayer and MacCreight (2) suggest that by using this method of insertion, the utilization of the tree will be approximately 66-70% of the total available space in the tree. The term utilization as used here is the ratio of the number of keys actually in the tree to the maximum number of keys the tree can hold. The utilization is given in term of a percentage and defined by the formula (9):

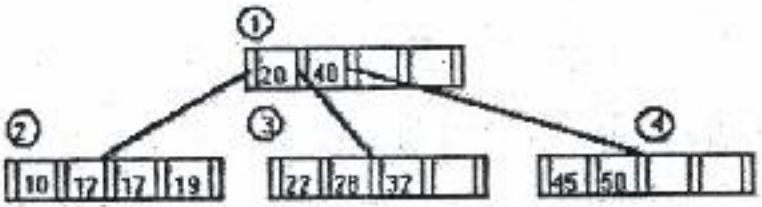
$$U = 100 * KEYS / (NODES * MAX)$$

where

KEYS Is the number of keys in the tree
NODES is the number of nodes in the tree
MAX is the maximum keys per node (one less than the order of the tree)



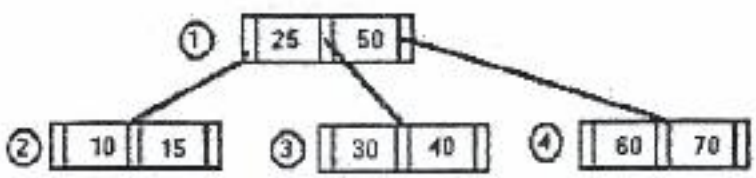
a) Before Insertion of Key 11



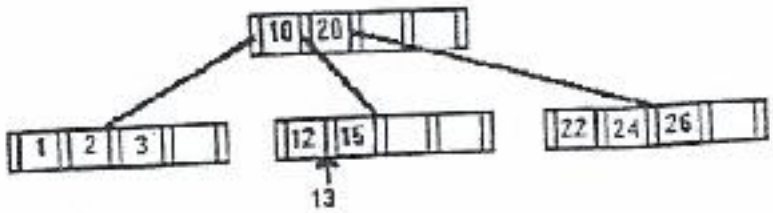
b) After Insertion of Key 11

Figure 6. An Order Five B-tree Illustrating the Two Way Split Process During Insertion

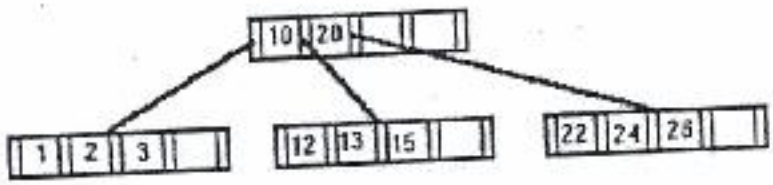
If the father node has overflowed because of this action, the entire two way split process is applied to the father node. Logically then, this splitting process may be propageted back up through the entire tree. If the node being split does not have a father then it must be the root node. In this case the two way split creates a new root node which necessarily adds one level to the height of the tree (Figure 7). The fact that the number of levels in the B-tree increases only when a new root node is added to the tree shows that a B-tree is actually built from the bottom up.



a) Before Insertion of Key 17



a) Before Insertion 13



b) After Insertion 13

Figure 5. An Order Five B-Tree Illustrating the Basic Insertion Process

The nodes in a B-tree have a maximum (and also a minimum) size which, if exceeded during the insertion of a key, will violate one of the basic properties of B-trees. If this occurs then a corrective action known as two way splitting must take place. A two way split constitutes taking an "overfull node" and dividing it into two nodes, each of which is approximately one-half full. Due to the fact that this adds one new node to the tree, a new pointer must be created in as much as all nodes in a tree except the root must be pointed to by another node. Therefore the middle key from the node being split is moved up into the father of this node (the predecessor node of any node in the tree). This key is inserted into the father node in its correct ordered position and all other keys and pointers are shifted to the right one place. An empty pointer space is created which can now be used to point to the new node just added to the tree due to the split (Figure 6).

If a search for key 25 is made in the tree in Figure 4, then it is immediately found in position 1 of node 1.

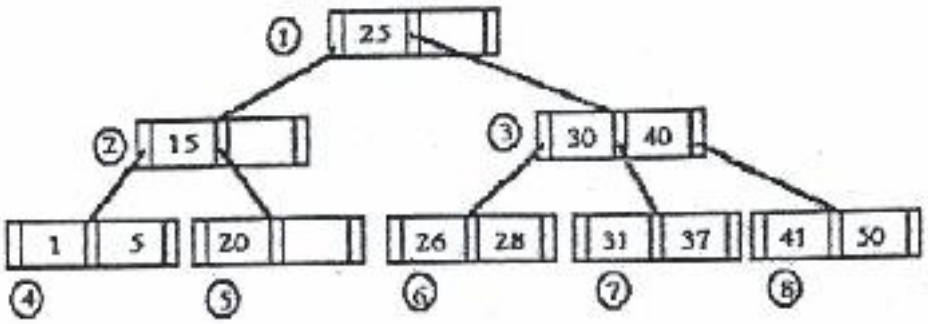


Figure 4. An Order Three B_Tree With Three Levels

Using the same tree, if a search is made for the key 20, then the following action would take place: the root node 1 is scanned and it is determined that the key is less than 25, therefore the left branch is taken and node 2 is accessed. The key is larger than the single key in node 2 so the right branch is taken and 5 is accessed. The key is finally located in position 1 of node 5. One final example is presented to illustrate the method used in determining that a key is absent from the tree. When searching for key 25 in Figure 4, the root node is scanned, the right branch is selected, and node 3 is accessed. The key falls between the first and second keys in node 3 so pointer 2 is used to access node 7. It is important to remember that as soon as one key in a node is found to be larger than the key sought, the scan can be stopped because all keys further to the right will also be larger. Scanning across node 7, it is determined that the key falls between the first and second keys in the node implying that the pointer 1 is to be selected for the next branch. Since this pointer is null, it is determined that the key is not in the tree but should be in position 2 of node 7. From this example, it is evident that a key cannot be determined to be absent from the tree until one node from each level has been inspected. These examples also show that any new key that will be added to the tree will be inserted into a leaf node.

2.2 Insertion

Insertion of a new keys into a B-tree is somewhat different from insertion into an ordinary binary tree or an indexed sequential file because of the special properties of B-trees. By using the B-tree search technique, the position within the tree in which the key should be found may be determined. If, while searching for this position, the key is found to be already in the tree, then some type of error condition should be raised. If the key is not found, then it should be inserted in the correct ordered position in the appropriate leaf node (Figure 5)

2. OPERATIONS ON B_TREES

In order to satisfy the five properties of B-trees as previously stated, some rigid rules must be given regarding the three basic operations to be performed on B-trees. Before discussing these operations (search, insert, and delete), it is necessary to mention that there are basically three different classes of B-trees:

[a] Those that hold information only in the leaf nodes.
[b] Those that hold information in all levels of the tree.
[c] Thosethat hold only pointers to the records which are stored elsewhere.

In the first class of B-tree, all levels except the bottom level of the tree contain only keys and are used as multilevel file index, similar to the indexed sequential structure. This paper will not be concerened with this type of structure.

B-tree of the seconed class contain information at all levels. The records are in the tree adjacent to their associated keys

The third class contain nothing but pointers to the records which are stored external to the tree.

2.1 Search

The operation of searching for a record is the basis of all operations on a B-tree (as it is for any file structure) because all other operations depend on it. A nonsequential search for a key will always begin with the root node. For this reason it might be advisable to keep the current root node in the interanal memory of the computer at all times. When searching any B-tree node, either a linear or binary search may be used since the keys in a node are stored in ascending order from left to right. If the key is found when searching a node, then the related information that needs to be "retained" is the identification of the node in which the key is found and its position within this node. If the key is found in this node, then it is easy to determine between which two keys this key should be logically located. If the pointer of the node at this point is not the null pointer, then the successor of this node is accessed and the search process starts over as if this were the new root node (this is actually the root node for a subtree). This process continues until the key is found or the bottom of the tree is reached.

If this pointer is null however, the node is a leaf node (the node is on the bottom level of the tree) and the related information that should be retained in this case is the identification of the node the key should be in and the position which the key should occupy within this node.

An example of a tree with № = 2,000,000 and conveniently chosen order of m = 200, the maximum number of levels and thus the maximum number of probes into the B-tree will be four. (It should be recognized that the number of node probes is very important if each probe requires a reference to a secondary storage. However, if a large number of keys are contained within a node, search performance within the node itself is also a factor).

As can be seen by equation (1), the number of levels and thus the maximum number of probes into a B-tree depends not only on the number of keys in the tree but also upon the order of the B-tree. For this reason, somthing should be said about the selection of the order when designing a B-tree. The trade-off is between the number of levels in the tree and the size of the nodes. If the tree is stored on a secondary memory device such as disc, an ideal node size is the same as the capacity of a track. The tree can be completely contained within the main memory (if storage capacity permits) of the computer to eliminate the extra delay in access time due to the disc arm movement and disc rotation. But for extremely large trees this could be impossible. If the primary memory of the computer is actually a virtual storage environment then a page of the machine's memory could conveniently contain one node of the B-tree.

There are three distinct methods of constructing B-trees to organize information. A record associated with a key may be stored adjacent to the key in the tree; all records may be stored in the lowest level of the tree; and finally the records may be stored in a manner entirely independent of the tree structure. The primary application of this type of B-tree is for files that are internally structured and stored such as symbol tables.

The second type of B-tree structure incorporates the concept of storing records directly in the tree but only at the lowest level. Here the keys in the upper levels of the tree act as a set of indices to control the traversal down to the correct "block" which contains the traget record. For this reason this type of tree structure is quite similar to indexed sequential. IBM's Virtual Storage Access Method (VSAM) is an example of an application of this type of B-tree.

In the third type of B-tree structure, the key is used solely as an index. The records themselves are not stored directly in the tree but are stored separately and are accessed by pointers in the tree. Thus, there is no need for the records to be physically ordered. This type of file structure is well suited to random access requirements. In this method of structuring B-tree, unlike the previous two, all nodes will hold the same type of information.

Beside the three methods of structuring B-tree mentioned above, it should be evident that they may be used in a variety of applications. In fact B-trees are well suited for many applications that involve the use of files or tables of information that must be randomly accessable via a key. Algorithms are available for maintaing B-tree balanced during the insertion and deletion (1).

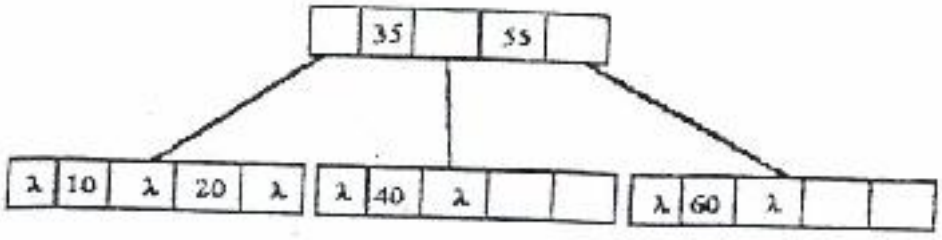


Figure 3. An Order Three B-Tree

B trees can have a variable number of levels as did binary trees and multi-branching from the nodes as did indexed sequential; but these alone could be properties of any multi-branching tree (a tree with a variable number of branches from each node). The characteristics of B-trees which make them unique are follows(1):

[a] Every node has at most m sons.

[b] Every node, except the root and the leaves has at least $\lceil m/2 \rceil$ sons (the symbol $\lceil x \rceil$ indicates the smallest integer that is greater than or equal to x).

[c] The root node has at least two sons unless it is a leaf, in which case it is the only node in the tree (as in Figure 2).

[d] All leaves will have null pointers and will be on the same level, which in fact will be the bottom level of the tree.

[e] A non-leaf node with k sons has k-1 keys.

This property along with the first two implies that every node, except the root, will contain between $\lceil m/2 - 1 \rceil$ and m-1 keys.

A B-tree has the important characteristics of being built from the bottom up. This implies that when a new level is added to the tree, it will be added as a new root node. Because of this method of building a B-tree, in contrast to a binary tree which is built from the top down, the B-tree constantly stays in balance; all leaf nodes are on the bottom level and all keys in the leaf nodes may be reached by the same number of probes into the tree. Since this is a balanced tree, the longest possible search will be equal to the number of levels in the B-tree. Knuth (4) has given an upper bounded on the number of levels (l) in a B-tree of order m with N keys to be

$$l <= 1 + \log_{\lceil m/2 \rceil}((N+1)/2) \qquad (1)$$

1. B-Tree

Basically, a B-tree is a balanced tree with a prudetermined maximum number of branches from each node. This maximum number of branches (known as the "order" of the tree) can be any value greater than two. An illustration of a single B-tree node is given in Figure 1.

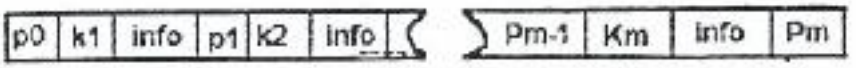


Figure 1. A Sample of B-tree Node

In this node, k_i represent the i-th key and INFO is the information (or possibly a pointer to the information) associated with this key. It should be noted here that the keys are in ascending order within any particular node. P_{i-1} is a field that points to a node or group of nodes containing keys less than k_i and p_i points to a node or group of nodes which contains keys greater than k_i. Depending on stored position of the B-tree, these pointers may be either addresses in primary memory or secondary storage.

A B-tree which is made up of only one node shown in Figure 2.

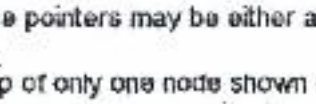


Figure 2. A B-tree Consisting of a Single Node

Notice that the tree is made up of one node which contains two keys and three pointers which are null (indicated by λ). Two important facts that may be pointed out about this B tree are that there is one level in the tree and the single node of the tree is also the root.

Figure 3 illustrates a little more complicated B-tree of order 3 (a maximum of three way branching from each node). This B-tree consists of two levels and contains four nodes. The single node on the top level (the node with keys of 35 and 55 in Figure 3) is the root node (as will be the case for all B-trees). The three nodes on the bottom level are leaf nodes due to the fact that they have null pointers (as shown in Figure 3, the nodes needs not be completely full).