

Diyala University
College of education for pure Science
Department of computer Science
Lecturer of Data Structure

Lecturer 1

المقدمة

تستخدم أنظمة الحواسيب بصورة أساسية في معالجة البيانات وعليه فإن طريقة تنظيم هذه البيانات تؤثر على سرعة المعالجة حيث أن سرعة المعالجة للبيانات والحصول على النتائج يعتبر من المتطلبات الأساسية في أنظمة الحواسيب ، وأيضاً تنظم هذه البيانات يؤثر على مدى استقلال الذّارة الرئيسية في أنظمة الحاسوب يجب أن يكون مرتفع إلى حد ما

المعلومات والبيانات:-

تعرف البيانات على أنها مجموعة من الحقائق والأفكار التي لم يتم معالجتها وتعرف بأنها المادة الخام للمعلومات ، وحتى تصبح البيانات معلومات لابد أن تعالج هذه البيانات بطريقة معينة لتعطي بيانات ذات فائدة معينة تسمى بالمعلومات ، تمتاز المعلومات عن البيانات في أنها مرتبة ، مصنفة ملخص وذات فائدة معينة.

من أهم العمليات التي يمكن استخدامها لمعالجة البيانات مايلي:-

- ١ -تجميع البيانات من مصادرها المختلفة.
 - ٢ -التحقق من صحة البيانات.
 - ٣ -ترميز البيانات والتأكد من صحة الترميز.
 - ٤ -تخزين البيانات على وسائط تخزين تمتاز بالسعة العالية وبسهولة استرجاع البيانات منها وسرعة الوصول إليها.
 - ٥ -ترتيب وتنظيم البيانات في هياكل بيانات محددة لتسهيل عملية فهم ودراسة البيانات.
 - ٦ -العمليات الحسابية والمنطقية التي تتم على البيانات باستخدام العلاقات المختلفة.
 - ٧ -فرز أو دمج البيانات معاً.
 - ٨ -آتابة التقارير المختلفة من البيانات التي تمت معالجتها.
- وأثر ما يهمنا هنا تنظيم البيانات في هياكل بيانات محددة . فماذا نقصد بهياكل البيانات ؟ ما هي أنواع هياكل آل البيانات ؟

الهيكلية خاصة من خصائص البيانات:-

دائماً ما نجد أن الأشياء تمتاز بمجموعة صفات ، وهذه الصفات تعتبر الخصائص المميزة والمحددة للأشياء ، وتتنظم هذه الصفات طبيعياً بشكل بنائي منظم.

حيث يعبر الجد عن الجذر "tree" . فمثلاً أن أسماء أفراد العائلة تنتظم تحت بناء يشبه الشجرة فيها والأبناء عن الفروع وأذلك الأحفاد وهكذا. وأمثال آخر ينتظم الناس في طابور لشراء حاجة ما يعبر الواقف أولاً عن رأس الطابور والأخير عن ذيل الطابور ودليل التلفون وغيرها من الأمثلة. آل هذه الأمثلة تبين لنا حقيقة واحدة ، هي أن المعلومات تمتاز بالترتيب هذا الترتيب يعرف "data structure". بالهيكل أو هياكل البيانات

ومن هنا نلخص الى تعريف هيكل البيانات ، حيث أنه تشكيل منظم لمجموعة من البيانات التي تشترك بصفة أو أثر فيما بينها وذلك لتؤدي غرضاً محدداً حول شيء أو مجموعة أشياء محددة. ويمكن أن نجد هذا التشكيل المنظم في الأمثلة التي سقناها سابقاً فمثلاً ، أن دليل التلفون يضم في آل صفحة من صفحاته عمودين أحدهما يدل على أسماء أشخاص معينين ، أو مؤسسات أو هيئات محددة والعود الأخر يدل على أرقام الهواتف التي يملكها الأفراد أو المؤسسات أو الهيئات الموجودة في المعود الأول. ويكتمل هذا التشكيل المنظم بترتيب العمود الأول أبجدياً وذلك لتسهيل عملية البحث عن فرد أو مؤسسة أو هيئة معينة.

إن هيكل البيانات ضروري لمعالجة البيانات نظراً لما تتمتع به هياكل البيانات من مميزات تساعد في الإسراع في الوصول إلى البيانات ومعالجتها ، فمثلاً استخدام المؤشرات في هياكل البيانات يساعد في عملية الحذف والإضافة ، فحذف عنصر معين من الهيكل لا يؤدي إلى إعادة ترتيب العناصر الأخرى ويكفي إجراء عملية التعديل على المؤشر وأذلك الحال بالنسبة للإضافة ، وبمعنى آخر يمكن حفظ عناصر البيانات للهيكل الواحد في أماكن مختلفة مكتفين فقط بإضافة مؤشر للعنصر بحيث يشير هذا المؤشر إلى موقع العنصر التابع في الهيكل مما يؤدي إلى الإسراع في تنفيذ آفة عمليات المعالجة لهذه العناصر ، وسوف نتناول لاحقاً هذه الأشياء بإسهاب. أما أسلفنا فإن عملية تجميع البيانات في هياكل مختلفة تؤدي إلى الإسراع في عمليات المعالجة فإن طريقة التخزين تلعب هي الأخرى دوراً هاماً في زيادة سرعة المعالجة لهذا لا بد من الإشارة إلى الترتيب الفيزيائي والترتيب المنطقي للبيانات. فالمعالجة البيانات فانه يتطلب وجود برنامج خاص بالمعالجة يجب أن يكون موجود في الذاكرة الرئيسية أما البيانات فيمكن أن توجد على وسط تخزين خارجي " وحدة إدخال " يتم نقلها جزءاً إلى الذاكرة الرئيسية بقراءتها ومن ثم معالجتها.

وأما هو معروف فإن الذاكرة الرئيسية تنقسم إلى خلايا معنوية وتعتمد سرعة الوصول إلى المعلومات على طريقة العنوان المستخدمة ؛ أما بالنسبة للعامل الثاني فإن سرعة الوصول إلى معلومات لإحضارها فتعتمد على سرعة الوسط المخزنة عليه ، أما ما يهمنا هنا هو التقليل من وقت تبادل المعلومات بين الذاكرة الرئيسية ووحدات الإدخال والإخراج ولهذا لا بد من الإشارة إلى الترتيب الفيزيائي والمنطقي للبيانات ، حيث يبين الترتيب الفيزيائي الشكل المخزنة به البيانات على وحدات التخزين الثانوية ويسمى هذا الشكل بالوحدة الفيزيائية أما وحدة البيانات " الهيكل مثلاً " فتسمى بالوحدة المنطقية ، وسوف نوضح هذا من خلال المثال التالي:

لنفرض أن ملف الطلاب يتكون من ١٠٠٠ سجل وأن معدل التكتل يساوي ٥) عدد السجلات في الكتلة الواحدة = ٥ سجلات (على فرض أن الملف مخزن على قرص مغنطيسي . لهذا ينظر إلى فينظر إليه أوحدة " block " على أنها وحدات فيزيائية أما السجل داخل الكتل " blocks " الكتل منطقية.

واستخدام الوحدات الفيزيائية في المعالجة يؤدي إلى زيادة السرعة في المعالجة وذلك بالتقليل من عمليات القراءة فيدلاً من قراءة خمسة سجلات " وحدات منطقية " يتم قراءة وحدة فيزيائية واحدة حيث تنتقل هذه الوحدة الفيزيائية مرة واحدة إلى الذاكرة الرئيسية وتخزن هناك في مناطق حيث يتم نقل وتبادل المعلومات إلى داخل الذاكرة " buffer " تسمى بمناطق التخزين المؤقتة الرئيسية و عملية التبادل هنا تأخذ وقت أقل.

ولزيادة سرعة المعالجة يمكن تنظيم طابور " queue " من مناطق التخزين المؤقتة " queue buffer "

إذا كان حجم الذاكرة يسمح بذلك.

Lecturer (2)

أنواع هياكل البيانات:-

تنقسم هياكل البيانات إلى نوعين محددين هما ، هياكل بيانات ثابتة " Static data structure " وهياكل متغيرة " Dynamic data structure " وذلك تبعاً لثبات حجم البيانات أو تغيرها

أما هياكل البيانات الثابتة فهي تلك المجموعة من الهياكل التي تحتوي على عدد من العناصر المحددة ، بحيث لا يزيد هذا العدد بالإضافة ولا ينقص بالحذف حيث أن ألا العمليتين غير مسموح بهما وتمتاز بعيوب عدة أهمها:-

- ضرورة توفير العدد اللازم والمتتابع من مواقع الذآرة لتخزين عناصر الهيكل.
- عدد المواقع المخصصة للهيكل ثابتة ولا تتغير .
- سرعة المعالجة بطيئة وخاصة سرعة تنفيذ عمليات الإضافة والحذف والتي تعتبر من أكثر العمليات المنفذة على هيكل البيانات حيث يصاحب تنفيذ هذه العمليات تنفيذ عمليات التحريك اللازمة للبيانات.

بينما نجد في هياكل البيانات المتغيرة تغييراً في حجمها وذلك لإمكانية الإضافة والحذف فيها ، مع تطور أنظمة الحاسوب أصبح بالإمكان معالجة عناصر الهيكل باستخدام المؤشرات " Pointers "

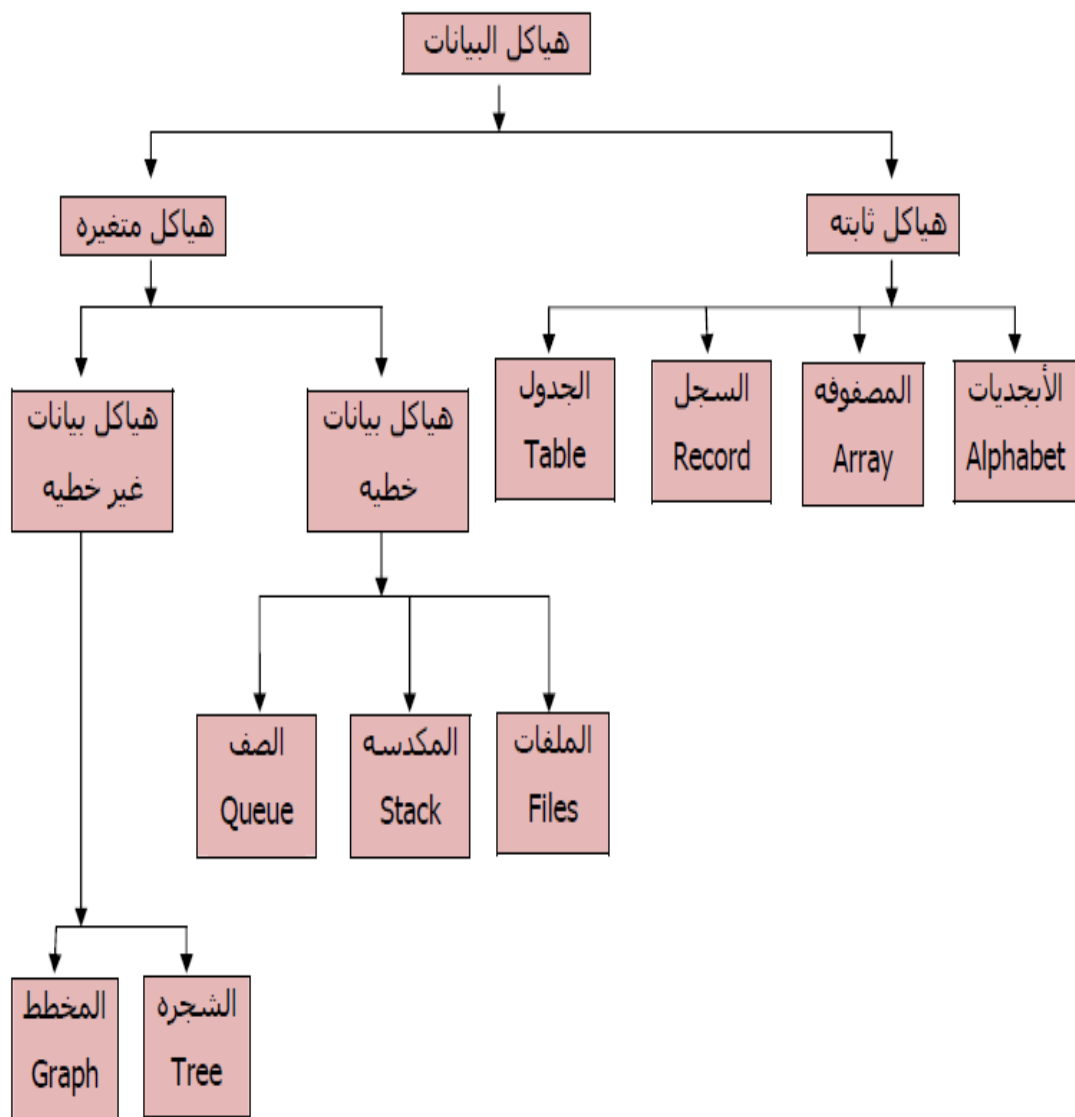
وبالتالي ظهرت مفاهيم هياكل البيانات الديناميكية ، إن استخدام هذه الهياكل لتنظيم البيانات أدى إلى التخلص من الكثير من العيوب المصاحبة لإستخدام الهياكل الثابتة وتوفر الهياكل الديناميكية:-

- سرعة معالجة عالية .
- استغلال الذآرة بشكل أفضل إذ ليس بالضرورة توفير مواقع متتابعة في الذآرة الرئيسية لتخزين عناصر الهيكل.
- حجم الهيكل الديناميكي غير ثابت ويعتمد فقط على حجم الذآرة المتوفر .

تستخدم في عملية تنظيم البيانات هياكل مختلفة " سواء آآنت ثابتة أو ديناميكية " ولكل هيكل من هذه الهياكل تطبيقاته المختلفة وطرق معالجه خاصة به ، وعليه سوف نطلع القارئ في هذا الكتاب على هياكل البيانات المختلفة مركزين بذلك على :-

- مفهوم الهيكل .
- خوارزميات عمليات معالجة الهيكل وتنفيذها .
- تطبيقات الهيكل.

الشكل التالي يبين تقسيمات هياكل البيانات المختلفة



Lecturer (3)

المصفوفات (Arrays)

هي عبارة عن هيكل من هياكل البيانات ، وهي عبارة عن مجموعة منتهية ومرتبطة من العناصر المتجانسة. والبيانات التي تخزن داخل المصفوفة لها شروط منها:-

١. أن تكون منتهية ، أي أن يكون للمصفوفة حجم معين والبيانات تكون محدده وإذا أردنا أن نتعامل مع المصفوفات فلا بد من تحديد حجم المصفوفة ويتم ذلك بواسطة المستخدم ، ومن سلبياتها أنه إذا تم حجز مساحة معينة للمصفوفة ولم تستغل هذه المساحة بشكل كامل فلا يمكن استغلال هذه المساحة بواسطة
٢. أن البيانات تكون مرتبة مثل $x_1, x_2, x_3 \dots$.
٣. أن تكون المصفوفة متجانسة ، ونقصد بالتجانس أنه لا يمكن جمع بيانات مختلفة في أنواعها وجعلها في مصفوفة واحدة ، وبالتالي لا بد أن تكون البيانات من نوع واحد مثل Integer, real, string, ...

نعني بقولنا منتهية (Finite) أي أن لها عدد محدود من العناصر ، ومرتبطة (ordered) أي أن عناصرها مرتبة الأول فالثاني فالثالث ... وهكذا ، ومتجانسة (homogenous) أي أن جميع عناصرها من نفس النوع (Type) .

أنواع المصفوفات

١. مصفوفة أحادية الأبعاد.
٢. مصفوفة ثنائية الأبعاد . وهما الأثر استخداماً (١ . ٢) .
٣. مصفوفة متعددة الأبعاد (Multidimensional array) .

الصيغة Syntax :-

هنالك طريقتان للتصريح عن المصفوفات:-

١. أن تقوم بتعريف المصفوفة على أنها نوع بيانات Data type فلا بد أن يتم حجز مساحة لها داخل الذاكرة ومن ثم نقوم بتعريف هذه البيانات (float , char, Integer) ، ويمكن أن تكون هنالك بيانات لا يوجد تعريف لها ولكن هنالك مواقع محددة في شكل البرنامج العام.
٢. تستخدم هذه الطريقة الثانية إذا كان هنالك مصفوفة واحدة ففي هذه الحالة أعرفها في منطقة التعريف وكذلك تستخدم هذه الطريقة إذا كان هنالك عدد من المصفوفات غير المتشابهة .

```
Int A[4][4];
```

أنواع المصفوفات:-

١. المصفوفة أحادية الأبعاد

- سنتعرف على كيفية تخزين البيانات داخل المصفوفة وكيفية الوصول إلى البيانات لإجراء عمليات المعالجة ومن ثم كتابة عدد من البرامج.
- إذا أردت أن أأخذ Data في المصفوفة أو أخرج Data استخدم For loop وان عملية التخزين في المصفوفة عملية قائمة بذاتها وعملية الاسترجاع في المصفوفة عملية قائمة بذاتها فبذلك نحتاج حلقة للتخزين وحلقة للاسترجاع ، وبعد المصفوفة هو الذي يحدد عدد الحلقات ، وفي المصفوفة أحادية الأبعاد أستخدم حلقة واحدة للتخزين وحلقة ثانية للإخراج .

Simple Example:-

```
program arrays;
var x:array[1..5]of integer;
i:integer;
begin
writeln('enter the data of array x....');
for i:=1 to 5 do
begin
write('enter element x['i,']: ');
readln(x[i]);
end;
writeln('the data of array x is:');
for i:=1 to 5 do
write(x[i], ' ');
readln;
end.
```

Lecturer (4)

المصفوفة ثنائية الأبعاد:-

وهي نوع من المصفوفات الأكثر استخداما ولها بعدان (2-Dimension) الاول يمثل عدد صفوف المصفوفة والثاني يمثل عدد اعمدة المصفوفة.

وتعرف المصفوفة بلغة C++ كما موضح ادناه

Int B[4][3]:

حيث من خلال هذا التعريف تم الاعلان عن مصفوفة اسمها B عدد صفوفها (4) صفوف وعدد اعمدتها (3) اعمدة.

ويتم قراءة وطباعة المصفوفة عن طريق (Double loop)

Example:-

program three;

type

matrix=array[1..30,1..30] of integer;

var x,y,z:matrix;

i,j,k,m1,m2,n1,n2:integer;

begin

writeln('enter the demintion of array x:','m1<=30,n1<=30');

readln(m1,n1);

writeln('enter the data of array x:');

for i:=1 to m1 do

for j:=1 to n1 do

begin

```

write('enter element x',i,j,' ');

readln(x[i,j]);

end;

writeln('enter the demintion of array y:', 'm2<=30,n2<=30');

readln(m2,n2);

writeln('enter the data of array y:');

for i:=1 to m2 do

for j:=1 to n2 do

begin

write('enter element y',i,j,' ');

readln(y[i,j]);

end;

if n1=m2 then

begin

for i:=1 to m1 do

for j:=1 to n2 do

begin

z[i,j]:=0;

for k:=1 to m2 do

z[i,j]:=z[i,j]+x[i,k]*y[k,j];

end;

writeln;

writeln('the data of array x is:');

```

```
for i:=1 to m1 do

begin

for j:=1 to n1 do

write(x[i,j], ' ');

writeln;

end;

writeln;

writeln('the data of array y is:');

for i:=1 to m2 do

begin

for j:=1 to n2 do

write(y[i,j], ' ');

writeln;

end;

writeln;

writeln('the data of array z is:');

for i:=1 to m1 do

begin

for j:=1 to n2 do

write(z[i,j], ' ');

writeln;

end;

end;
```

readln;

end.