المحاضرة 20 والأخيرة

◄ دكتور المادة: سمير جعفر

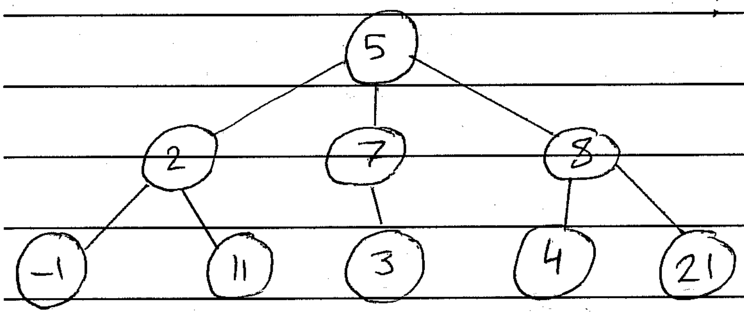
◄ عنوان المحاضرة: متابعة الأشجار + البيان

نظري ☑ عملي ☐

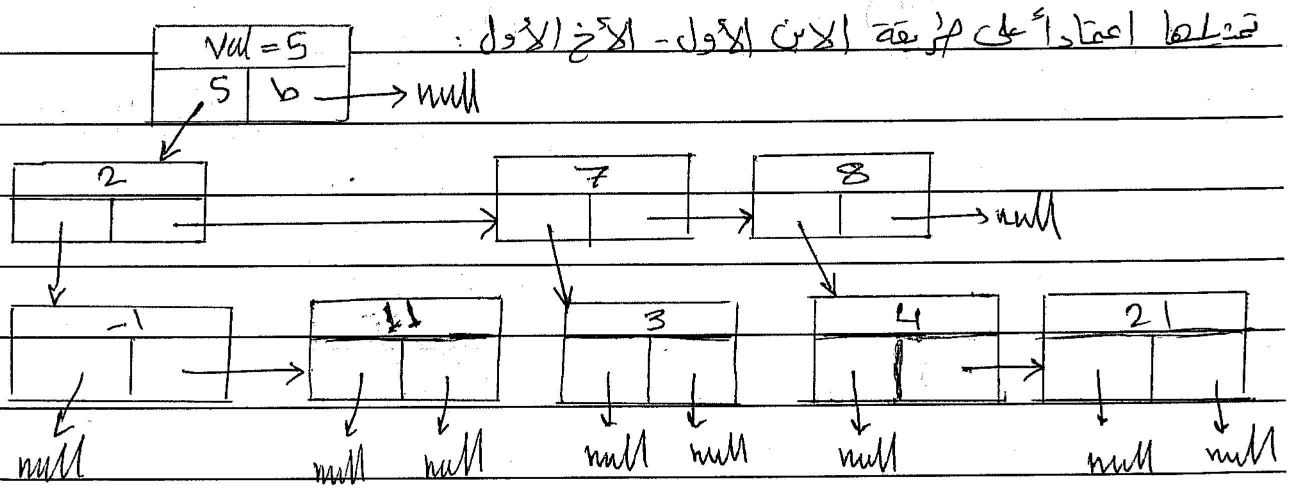
رأينا في المحاضرة السابقة كيف مثلنا الشجرة المعممة بالاعتماد على متجهة للأبناء، لكن هذه الطريقة تتطلب تحديد عدد أعظمي للأبناء، أما إذا كان عدد الأبناء غير محدود فسوف نستخدم طريقة جديدة نمثل فيها الشجرة عن طريق حقلين كلٌّ منهما من نوع الصف نفسه، الحقل الأول يدل على الابن الأول للعقدة، والحقل الثاني يدل على الأخ الأول للعقدة، فيكون الصف بالشكل:

```
class Gt {
  int val;
  Gt s;
  Gt b; }
```

مثال للتوضيح: لتكن لدينا الشجرة:



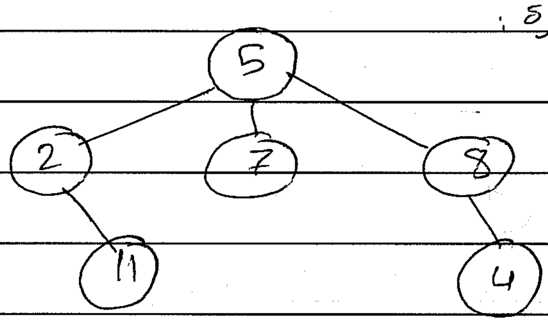
نمثلها اعتماداً على طريقة الابن الأول - الأخ الأول:



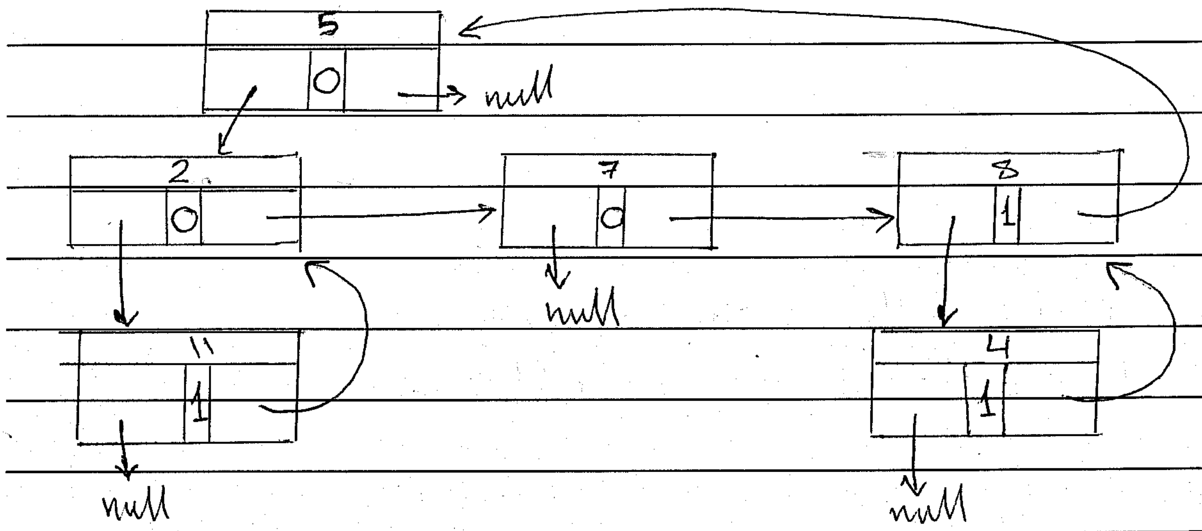
لكن في الحالة السابقة لا يمكننا معرفة الأب لكل عقدة.
لمعرفة الأب لكل عقدة سنجعل الأخ الأخير يشير بحقل الأخ الخاص به إلى الأب
(بدلاً من أن يشير إلى null) ، وهنا ظهرت مشكلة جديدة وهي كيف بإمكاننا أن نعرف
إذا كانت العقدة التي تؤشر عليها أب أم أخ ؟
ولحل هذه المشكلة سنضيف حقلاً إلى كل عقدة يميز إن كانت تشير إلى أب أم أخ ،
يمكننا جعل هذا الحقل عدداً صحيحاً int بحيث يأخذ إحدى القيمتين (0) أو (1) ،
أو يمكن تعريف متحول بولياني فيأخذ إحدى القيمتين (false) أو (true) .
في مثالنا سنضيف عدداً صحيحاً ونفرض أن القيمة (1) تدل إلى أن العقدة تشير إلى أب
والقيمة (0) تدل على أنها تشير إلى أخ . ((وعقدة الجذر سنعطيها القيمة (0) كذلك))
عندها سيصبح تعريف الصف بالشكل :

```
class Gt {
  int val;
  Gt s;
  Gt b;
  int x; }
```

مثال للتوضيح : إذا كانت لدينا الشجرة :



سنمثلها بالشكل :



- سنقوم بكتابة خوارزمية لطباعة أوراق شجرة معممة ممثلة بطريقة الابن الأول - الأخ الأول:

```
void printleaf(Gt t){
if(t != null){
    if(t.s == null) System.out.print(t.val);
    else printleaf(t.s);

 printleaf(t.b); }
                 }
```

الخوارزمية السابقة تتحقق أولاً من وجود العقدة ، ثم تتحقق من كون ابن هذه العقدة يؤشر على null ، فإذا كان كذلك تكون العقدة ورقة فنقوم بطباعة قيمتها وإلا (أي إذا كان للعقدة ابن) لا تكون ورقة فنستدعي الدالة من أجل الابن (حتى نتحقق من كون الابن ورقة)

وفي كلا الحالتين نريد التحقق إذا كان أخ العقدة ورقة ، لذلك قمنا باستدعاء الدالة من أجل العقدة الأخ خارج تعليمة (if ... else) .

- اكتب خوارزمية تقوم بحساب ارتفاع شجرة ثنائية معطاة . Bt

ملاحظة : ارتفاع الشجرة هو أكبر ارتفاع لعقدة منها ، أي عدد الوصلات من جذر الشجرة إلى أبعد عقدة عنه .

```
int h(Bt t){
① if(t==null) return 0;
② else return (1 + max(h(t.l) , h(t.r))) ;
}
```

① الشجرة غير موجودة فارتفاعها 0 .

② وإلا يرجع بشكل عودي 1 ((دلالة على الوصلة)) + القيمة الأكبر بين ارتفاعات العقد الأبناء اليمينية أو اليسارية .

ملاحظة أخيرة :

للتمييز بين أنواع الأشجار التي مرت معنا في المحاضرات السابقة :

شجرة ثنائية عادية	لا يوجد ترتيب على الأبناء	تحوي قيماً مكررة
شجرة بحث ثنائي	يوجد ترتيب على الأبناء	لا توجد قيم مكررة
شجرة معممة	لا يوجد ترتيب على الأبناء	تحوي قيماً مكررة

سنبدأ فيما يلي بالقسم الأخير من مقررنا والذي سنتحدث فيه عن النوع الثاني من بنى المعطيات غير الخطية وهو البيان ...

البيانات Graphs :

البيان $G(V,E)$ هو عبارة عن مجموعة غير خالية من الرؤوس V، ومجموعة من الأضلاع E

يكون لعناصرها الشكل : (v_1, v_2) ; $v_1, v_2 \in V$

لدينا نوعان من البيانات ← بيان موجّه Directed Graph

← بيان غير موجّه Undirected Graph

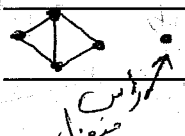
البيان التام: يحوي العدد الأعظمي من الأضلاع ((كل رأس فيه يتصل مع باقي الرؤوس))

- بيان تام عدد رؤوسه n ⟸ عدد أضلاعه = $\frac{n(n-1)}{2}$

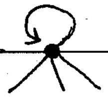
((وقد تناولنا هذه المفاهيم في مقرر نظرية البيان في السنة السابقة))

البيان المتماسك بقوة: هو بيان موجه وكل رأس فيه يتصل بباقي الرؤوس.

- بيان متماسك بقوة عدد رؤوسه n ⟸ عدد أضلاعه = $n(n-1)$



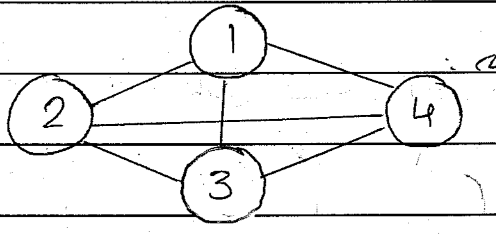
- الرأس المنعزل: رأس في البيان لا يتصل بأي من الرؤوس الأخرى



- الحلقة الذاتية self loop : رأس في بيان موجه يتصل بنفسه

أمثلة :

1) بيان غير موجّه :



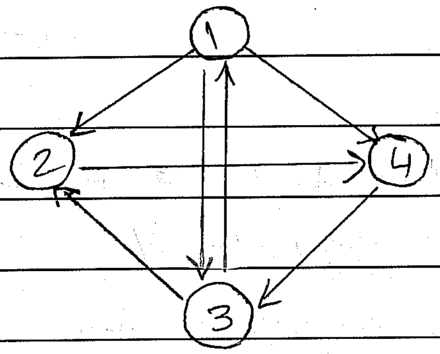
مجموعة الرؤوس : $V=\{1,2,3,4\}$

مجموعة أضلاعه : $E=\{(1,2),(2,3),(3,4),(1,4),(1,3),(2,4)\}$

حيث تكون الثنائية الدالة على الأضلاع تبديلية، أي مثلاً :

$(1,2) = (2,1)$

$\boxed{2}$ بيان موجه :



مجموعة أضلاعه : $E = \{ (1,2) , (1,4) , (1,3) , (2,4) , (3,2) , (3,1) , (4,3) \}$

ومنه : $(1,3) \neq (3,1)$

أحد أشكال تمثيل البيان برمجياً يكون عن طريق مصفوفة التجاور :

ننشئ مصفوفة مربعة من الأعداد الصحيحة من البعد n حيث n عدد رؤوس البيان.

ومن ثم إذا وجد ضلع يصل بين الرأس v_i و الرأس v_j فإننا نضع القيمة (1) في الخانة (i,j)، وإذا لم تكن العقدتان متصلتين نضع (0).

إذا كان البيان غير موجه تكون المصفوفة متناظرة (عناصر قطرها الرئيسي أصفار) وبذلك نضع نفس القيمة في الخانة (j,i)

ملاحظة : الشرح السابق هو في حال كان البيان غير موزون، أما إذا كان موزوناً (أي أضلاعه مزودة بقيم) فتكون قيم الأضلاع هي عناصر المصفوفة.

مثال التوضيح: في حالة البيان غير الموجه $\boxed{1}$ في المثال السابق تكون مصفوفة التجاور :

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 1 & 1 & 1 & 0 \end{array}$$

في حالة البيان الموجه [2] :

$$
\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 1 & 1 & 1 \\
2 & 0 & 0 & 0 & 1 \\
3 & 1 & 1 & 0 & 0 \\
4 & 0 & 0 & 1 & 0
\end{array}
$$

التجوال في البيان :

[1] التجوال في العمق أولاً « Depth first search » اختصاراً DFS

[2] التجوال في العرض أولاً « Breadth first search » اختصاراً BFS

ولنتعرّف على الطريقتين السابقتين ((الهدف منها المرور على جميع رؤوس البيان)) :

[1] هذا التجوال يتم بالانتقال من عقدة البداية إلى العمق (لآخر عقدة فيه))

حيث نختار عقدة بداية ثم مجاوراتها ، ثم مجاور العقدة الجديدة بشرط أثناء عند اختيار مجاور يجب أن نختار مجاوراً لم نمر عليه ، وهكذا حتى الوصول إلى عقدة ليس لها مجاورات عندها نرجع خطوة لمجاور لم نمر عليه ، وهكذا ....

في البيان الموجه في المثال السابق :

DFS = {1 , 2 , 4 , 3}

[2] تعتمد على مسح البيان بدوائر .

حيث نختار عقدة بداية ، ثم كل مجاوراتها ، ثم مجاورات أوّل عقدة من مجاورات عقدة البداية ((نقصد المجاورات التي لم نمر عليها مسبقاً))

وهكذا حتى يصبح رتل المجاورات فارغاً ((حتى نكون قد مررنا على جميع عقد البيان)).

في البيان الموجه في المثال السابق :

BFS = {1 , 2 , 3 , 4}

انتهت المحاضرة

تصحيح لبعض الأخطاء في المحاضرات السابقة :

- في المحاضرات الأولى يجب الانتباه عند كتابة البرامج إلى الأحرف الصغيرة والكبيرة.
الكلمات String , Stdin تبدأ بحرف كبير
والكلمة static تبدأ بحرف صغير.

- في المحاضرة الرابعة - الصفحة 2 :
عند تعريف متجهين : استخدام الشكل int[] A,B; ✓
صحيح ، والخطأ كتابة :
X int A,B[];
لأن هنا سيصرح عن متغير صحيح A ومتجه أعداد صحيحة B .

انتهى المقرر