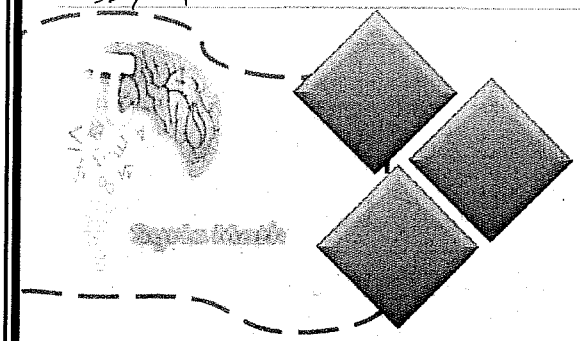


المحاضرة 18

نظري ☑
عملي ☐

◄ دكتور المادة: سمير جعفر

◄ عنوان المحاضرة: تعاريف في الأشجار
+ شجرة البحث الثنائي.

أهلاً بكم أصدقائي في المحاضرة الثامنة عشرة من مقررنا البرمجة والخوارزميات المتقدمة والتي سنتابع فيها الحديث عن بنى المعطيات ((بالقرب من الشجرة)).

الشجرة Tree: هي بنية معطيات غير خطية منظمة تنظيم هرمي، هذا يعني وجود عقدة أساسية ((عقدة بداية تسمى عقدة الجذر)) يمكن الانتقال منها إلى مجموعة جديدة من العقد، ويمكن الانتقال من عقدة من المجموعة الجديدة إلى مجموعة عقد جديدة وهكذا....

((الانتقال يتم عبر وصلة بين العقدتين))

- إذاً الشجرة مكونة من عقد Nodes ووصلات links.

أنواع الأشجار:

1- الشجرة الثنائية: هي شجرة لكل عقدة فيها ولدان على الأكثر.

2- الشجرة المرجة: هي شجرة لا قيد فيها على عدد أبناء العقدة ((يمكن أن توجد عقدة أكثر من ولدان)).

- والأشجار الثنائية حالات خاصة:

1- شجرة كاملة Complete: جميع مستوياتها ممتلئة تماماً بالعقد باستثناء الأوراق (المستوى الأخير).

2- شجرة تامة Perfect: جميع العقد الداخلية لها ولدان وجميع الأوراق في نفس المستوى.

3- شجرة خطية: لكل عقدة فيها ولد واحد على الأكثر.

تطلب ورقياً من المكتبة الجامعية

تعاريف عامة :

حجم الشجرة Size of tree : هو عدد عقدها.

عرض الشجرة : هو أكبر عدد عقد في مستوى.

ارتفاع عقدة : بعد العقدة عن الجذر (عدد الوصلات بينها وبين الجذر)

ارتفاع الشجرة : هو أكبر ارتفاع بين ارتفاعات الأوراق.

العقدة الداخلية : كل عقدة يخرج منها وصلة على الأقل. (أي كل عقدة لديها أبناء)

العقدة الخارجية (الورقة leaf) : عقدة ليس لها أبناء.

مسافة التجول الداخلي : مجموع ارتفاعات العقد الداخلية.

مسافة التجول الخارجي : مجموع ارتفاعات العقد الخارجية.

مسافة التجول : مجموع ارتفاعات جميع العقد.

شجرة البحث الثنائي Binary search tree

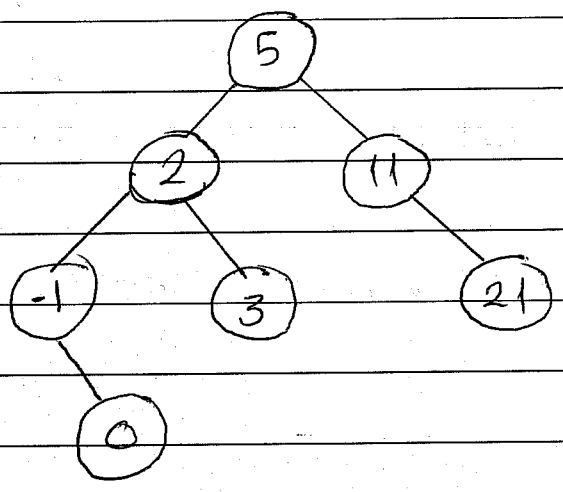
هي شجرة ثنائية تنظم عقدها وفق قانون معين بحيث تكون فيها قيمة الابن الأيسر أصغر من الأب، والابن الأيمن أكبر منه.

ملاحظة : شجرة البحث الثنائي لا تحوي قيماً مكررة.

مثال : لدينا القيم التالية :

$$\{5, 11, 2, -1, 0, 21, 3\}$$

نريد تمثيلها في شجرة بحث ثنائي، تكون الشجرة بالشكل :



(أول قيمة هي الجذر، ثم ننظر للقيم التي تليها بالترتيب: القيمة الأكبر تمثل الابن الأيمن والأصغر تمثل الابن الأيسر)

تطلب ورقياً من المكتبة الجامعية

تمثيل الشجرة بلغة Java :

كل عقدة من الشجرة تحوي بيانات ولها ولدان كل منها هو عقدة جديدة، إذاً سنقوم بإنشاء صف يمثل عقدة.

هذا الصف سيحوي : ① قيمة البيانات في العقدة و ② وصل الابن الأيمن r و ③ وصل الابن الأيسر l، كل منها سيكون من نوع الصف نفسه. (لأن كل منها هو عقدة كذلك، وسيحوي قيمة وسيكون أباً لعقدتين جديدتين، وهكذا حتى نصل إلى الأوراق فتوشر المؤشرات l, r على null، كون الأوراق ليس لها أبناء)

يكون الصف كما يلي:

```
class Node {
    int val;
    Node l;
    Node r;
```

(أبي الصف بأخذ قيمة a كوسيط

```
    Node (int a) { val = a; }   ← (val وتعطى هذه القيمة لـ
}
```

- إذا أردنا كتابة الشجرة في المثال السابق بلغة Java اعتماداً على الصف Node:

(أولاً نعرف عقدة من نوع الصف Node وسماها root لأنها تمثل الجذر)

```
Node root;          ← (تصريح عن الغرض root)
root = new Node(5);  ← (حجز وإعطاؤه القيمة 5 باستخدام الباني)
```

```
root.l = new Node(2);   ← (حجز وإعطاء القيمة 2 للابن الأيسر)
root.r = new Node(11);  ← (حجز وإعطاء القيمة 11 للابن الأيمن)
```

وتم ذلك عن طريق باني الصف أعلاه.

```
(root.l).l = new Node(-1);   (حجز وإعطاء القيمة -1 للابن
اليساري للعقدة الثانية (والتي هي الابن اليساري للجذر))
(root.l).r = new Node(3);   (حجز وإعطاء القيمة 3 للابن اليميني
للعقدة الثانية)
[...وهكذا];
```


تطلب ورقياً من المكتبة الجامعية
المكتبة الجامعية الحديثة الطالبية

- إذا كانت جميع المؤشرات تؤشر على null عندها نعلم أن العقدة هي ورقة.
((نعني هنا المؤشرات التي تدل على أبناء العقدة (ولد)))

التجول في شجرة ثنائية:

1- التجول بالترتيب السابق Pre-order: أب - يسار - يمين.
أي عند كل عقدة نبدأ بالأب ثم ننتقل إلى الابن الأيسر ثم الابن الأيمن.

2- التجول بالترتيب النظامي In-order: يسار - أب - يمين.
عند كل عقدة نبدأ من الابن الأيسر ثم ننتقل إلى الأب ثم الابن الأيمن.

3- التجول بالترتيب اللاحق Post-order: يسار - يمين - أب.
عند كل عقدة نبدأ من الابن الأيسر ثم ننتقل للابن الأيمن ثم العقدة الأب.

مثال: نسقط الطرق السابقة على الشجرة في المثال السابق:

Pre-order = < 5, 2, -1, 0, 3, 11, 21 >

حيث بدأنا من الجذر 5، ثم انتقلنا إلى ابنه اليساري 2، أصبحنا واقفين في العقدة 2 وهي أب لعقد أخرى، فنتقل إلى ابنها اليساري 1-، ونفس الطريقة إلى ابنه اليساري (هنا ليس له ابن يساري) لذلك ننتقل إلى الخطوة التالية وهي ابنه اليمين 0، وهو ورقة لا يملك أبناء، فنعود إلى الأب السابق 2 ونطبع ابنه اليمين 3، ثم نعود إلى الجذر 5 ونطبع ابنه اليمين 11، أصبحنا واقفين في العقدة 11 التي لا تملك ابن يساري فنتقل إلى ابنها اليمين 21، وبذلك نكون تجولنا في الشجرة كاملة.
الطريقتان التاليتان بنفس الطريقة:

In-order = < -1, 0, 2, 3, 5, 11, 21 >

Post-order = < 0, -1, 3, 2, 21, 11, 5 >

انتهت المحاضرة

تطلب ورقياً من المكتبة الجامعية