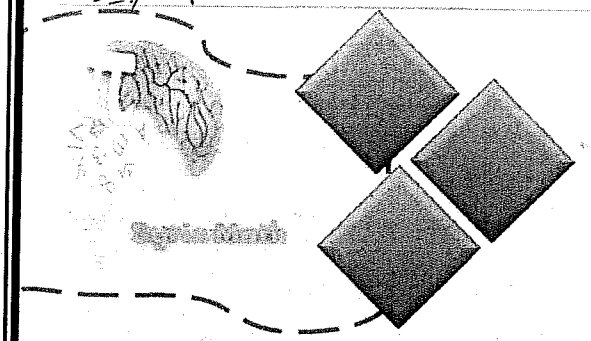

المحاضرة 15

نظري ☑
عملي ☐

◄ دكتور المادة: سمير جعفر

◄ عنوان المحاضرة: خوارزميات متعلقة بالأشجار

أهلاً بكم في المحاضرة التاسعة عشرة من مقررنا ، والتي سنتابع فيها الحديث عن الأشجار.

تمرين : لنكن لدينا شجرة بحث ثنائي BST ، والمطلوب :

اكتب خوارزمية تقوم البحث عن عنصر x فيما إذا كان موجوداً فيها أم لا.

ثم احسب كلفة الخوارزمية.

الحل : سنمرر العنصر الذي نريد البحث عنه كوسيط لدالة البحث ، بدايةً يجب النظر لقيمة الجذر إذا كانت مساوية لـ x فهو موجود في الشجرة ، وإذا لم تكن مساوية نتحقق من كونها أكبر أم أصغر من قيمة الجذر ، إذا كانت أكبر نتابع البحث في الشجرة الجزئية اليمنى ((أي ننتقل إلى الابن الأيمن)) ، وإذا كانت أصغر نتابع البحث في الشجرة الجزئية اليسارية ((ننتقل إلى الابن الأيسر)) ، ونتابع البحث بنفس الطريقة.

(وذلك اعتماداً على تعريف شجرة البحث الثنائي).

- الشرح السابق سيطبق على كل عقدة من الشجرة حتى نجد العنصر ، لذلك من الواضح أن الدالة يجب أن تكون عودية.

```
① boolean search (BST t, int x){
② if (t != null){
③ ←  if (t.val == x) return true;
     else if (x > t.val) return (search(t.r, x));
     else return (search(t.l, x)); }
④ else return false; }
```

① دالة اسمها search نوع إرجاعها بولياني ((true تدل على أن العنصر موجود ،

تطلب ورقياً من المكتبة الجامعية

و false على أنه غير موجود) ونأخذ وسيطين ، الشجرة والعنصر المراد البحث عنه.

(2) نتحقق من وجود الشجرة (أي أنها غير خالية)، فإذا كانت موجودة نتابع.

(3) إذا كان العنصر يساوي قيمة العقدة يرجع true (العنصر موجود)
وإلا يتحقق إذا كان أكبر منها فيبحث عنه يميناً ، وإلا يبحث يساراً.

(4) إذا كانت الشجرة خالية يرجع false.

- إيجاد الكلفة: بالنسبة لعمليات المقارنة:

في كل مرة نقوم بالبحث نتابع الخوارزمية إما القسم الأيمن من الشجرة أو الأيسر، أي أن عدد المقارنات التي نحتاج فيها للنزول إلى النصف في كل استدعاء.
فتكون الكلفة كما يلي:

$$T(n) = 1 + T\left(\frac{n}{2}\right)$$
$$= 1 + \left(1 + T\left(\frac{\frac{n}{2}}{2}\right)\right)$$
$$= 2 + T\left(\frac{n}{2^2}\right)$$
$$\vdots$$
$$= i + T\left(\frac{n}{2^i}\right)$$

شرط التوقف يكون عندما $\frac{n}{2^i} = 1$ ، أي عندما تبقى لدينا عقدة واحدة.

$$\frac{n}{2^i} = 1 \Rightarrow n = 2^i$$
$$\Rightarrow i = \log_2 n$$

بالتعويض:

$$\Rightarrow T(n) = \log_2 n + 1$$

$T(1)$ في الكلفة حال وجود عقدة واحدة ← لدينا عملية مقارنة واحدة.

تطلب ورقياً من المكتبة الجامعية

الشجرة العامة : في الشجرة العامة لا نعلم عدد الأبناء لكل عقدة و بالتالي لا نستطيع تمثيلها كما مثلنا الشجرة الثنائية ، لكن يمكننا وضع حد أعلى لعدد أبناء العقدة ونستخدم ضمن بنية العقدة (الصف الذي يمثل العقدة) متجهة عناصرها من نوع الصف لتمثل أبناء العقدة ، فيكون الصف كما يلي :

```
class Gt{
    int val;
    Gt[10] a; }
```

ملاحظة : وضعنا العدد الأعلى للمتجهة 10 ، أي وضعنا شرطاً أن عدد أبناء العقدة لن يتجاوز الـ 10 عقد .

- لنكتب الآن خوارزمية لطباعة أوراق شجرة معينة اعتماداً على الصف السابق :

```
void printleaf (Gt t){          ((void نوع إرجاعها))
① if (t != null){
② if (t.a[0] == null) System.out.print(t.val);
③ else for(int i=0; i<t.a.length; i++)
        printleaf(t.a[i]); } }
```

① نتحقق من شرط وجود الشجرة .

② تكون العقدة ورقة إذا لم يكن لها أبناء ، و أبناء كل عقدة ستكون مرتبة في المتجهة الخاصة بها a ، لذلك يكفي التحقق من أول عنصر في المتجهة a ، فإذا كان يشير إلى null ليس للعقدة أولاد فتكون ورقة ونطبع قيمتها .

③ إذا كان للعقدة أبناء (ليست ورقة) عرفنا حلقة for لنمر على أبناء العقدة ونستدعي التابع printleaf بالنسبة لكل من الأبناء لنتحقق من كونه ورقة أم لا .

انتهت المحاضرة .

WhatsApp : 0997378154 Facebook_Page : IOM
تطلب ورقياً من المكتبة الجامعية
الكلية الطبية الحديثة الطلابية

WhatsApp : 0997378154
Facebook_Page : IOM
تطلب ورقياً من المكتبة الجامعية
المكتبة الجامعية للطلاب