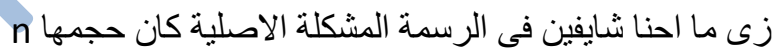


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الفكرة ان احنا عندنا مشكلة بتتقسم لعدد  $a$  من المشاكل الفرعية كل مشكلة فرعية حجمها بيساوي نسبة  $b$  من حجم المشكلة الاصلية



## طب بتنتھی امتی؟

بنتتهي لما يكون حجم المشكلة الفرعية اللي هو  $n/b = 1$  .... ليه؟

لان ده اصغر حجم للمشكلة عنده حل

طيب يبقى الوقت النهائي اللازم لحل المشكلة من اولها لآخرها كام؟

## هـنجمع الوقت بتاع كل مشكلة فرعية

Level 0 التكلفة بتاعته هي  $f(n)$  بس

Level 1 التكلفة بتاعته  $a.f(n/b)$

تمام ... بعد كده ال  $n/b$  هنتقسم تانى و تنقسم على  $b$  فتهبقى حجم المشكلة الجديد  $n/b^2$   
و كل مشكلة فرعية هيطلع منها عدد  $a$  من المشاكل الفرعية يبقى عدد المشكل كلها فى  
المستوى الجديد  $a^2$

و هكذا لغاية ما نوصل لآخر مستوى و هنسميه  $h$  هتكون التكلفة بتاعته

$$a^h \times f\left(\frac{n}{b^h}\right)$$

و بما ان عندنا ال  $h$  دى لما يكون حجم المشكلة = 1

$$\therefore \frac{n}{b^h} = 1$$

$$\therefore b^h = n$$

$$\therefore h = \log_b n$$

لو حبينا نحسب عدد المشاكل الفرعية الكلي

$$\text{total number of subproblems} = a^h = a^{\log_b n} = n^{\log_b a}$$

اما لو حبينا نحسب التكلفة الكلية فتهكون كالاتى تكلفة المشكلة كلها

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

و ال  $T(n/b)$  هتكون زيها

$$T(n) = a^2 \cdot T\left(\frac{n}{b^2}\right) + f(n)$$

فدى القوانين الاساسية فى ال recursion tree و اللى هنستخدمها بعد كده فى حساب  
كفاءة ال divide & conquer algorithms

مع جزيل الشكر للطالب حسن أحمد حسن على الرسومات

ارجو ان كل شي يكون واضح باذن الله

و اى استفسار على المنتدى ان شاء الله

تمت بحمد الله