Sylvain, Raphaël
111 124 564

Question 4

Analyse de `construire_noeud`

Notons que cette fonction possède un fonction privée récursive (`build_node`) et que la seule appel de la fonction `construire_noeud` est à appel à cette fonction privée. Nous évalurons donc la fonction `build_node`.

Taille de l'instance

La taille de l'instance est n, le nombre d'éléments compris entre l'index `beginIndex` et l'index `endIndex` inclusivement. ($n =$`endIndex`$-$`beginIndex`$+1$)

Opération de base

Nous allons choisir comme opération de base l'appel à la comparaison `nbPoints == 1`

Cet opération est l'opération la plus effectué dans la fonction, car il n'y a pas de boucle itérative suivant cet opération et qu'elle n'est pas imbriqué dans des conditionnels.

Le nombre d'opération est dépendant uniquement dans la valeur de n.

Analyse de la récusion

La récurence peut être donnée par la fonction suivante

$$C(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 1 + C(\lceil \frac{n}{2} \rceil) + C(\lfloor \frac{n}{2} \rfloor) & \text{si } n > 1 \end{cases}$$

Nous la modifirons ainsi :

$$\begin{aligned} & C\left(n\right) = \begin{cases} 1 & \text{si } n \leq 1 \\ 1 + C(\lceil \frac{n}{2} \rceil) + C(\lfloor \frac{n}{2} \rfloor) & \text{si } n > 1 \end{cases} \\ \Rightarrow \quad & \langle \text{ En supposant que } n = 2^k \text{ } \rangle \\ & C\left(2^k\right) = \begin{cases} 1 & \text{si } k = 0 \\ 1 + C(\left\lceil \frac{2^k}{2} \right\rceil) + C(\left\lfloor \frac{2^k}{2} \right\rfloor) & \text{si } k \geq 1 \end{cases} \\ \Rightarrow \quad & \langle \text{ Puisque } \tfrac{2^k}{2} \text{ est toujours un entier } \rangle \end{aligned}$$

$$C\left(2^k\right) = \begin{cases} 1 & \text{si } k = 0 \\ 1 + 2 \cdot C(\frac{2^k}{2}) & \text{si } k \geq 1 \end{cases}$$

$\Rightarrow$ $\langle$ Rammener sur n $\rangle$

$$C\left(n\right) = \begin{cases} 1 & \text{si } n \leq 1 \\ 1 + 2 \cdot C(\frac{n}{2}) & \text{si } n > 1 \wedge n = 2^k \forall k \in \mathbb{N}^+ \end{cases}$$

Nous avons ainsi une forme où le théorème général avec
$r = 2 \wedge b = 2 \wedge f\left(n\right) = n^0 \wedge d = 0$
Nous obtenons la forme 3 du théorème général
$r = 2 > 1 = 2^0 = b^d$
donc
$C\left(n\right) \in \Theta\left(n^{log_2 2}\right) \equiv C\left(n\right) \in \Theta\left(n\right)$

Conclusion

La fonction `construire_noeud` s'exécute donc en temps linéaire par rapport au nombre de points.