Énoncé

Considérons un algorithme dont le temps d'exécution $t(n)$ dans le pire des cas sur une entrée de taille n vérifie :

$$t(n) \leq t(\lceil n/5 \rceil) + t(\lceil 7n/10 + 9/2 \rceil) + \alpha n + \beta$$

où α est un réel positif et β un réel.

Montrer par la méthode de substitution que le temps d'exécution de l'algorithme est au plus linéaire.

Correction

Nous commençons par considérer la fonction T telle que $T(n)$ soit le temps d'exécution de l'algorithme sur une entrée de taille au plus n dans le pire des cas. La fonction T majore t, il suffit donc de montrer qu'elle est au plus linéaire. Nous montrons facilement que T vérifie la même relation de récurrence et est une fonction croissante.

Nous cherchons un réel c tel que pour tout entier n nous ayons $T(n) \leq cn$. Si cette inégalité est vraie jusqu'au rang n_0 alors pour tout entier n supérieur ou égal à n_0 nous avons :

$$T(n) \leq n(9c/10 + \alpha) + (13c/2 + \beta)$$

Pour prouver le résultat cherché il suffit donc de trouver un entier n_0, un réel u (par exemple compris entre 0 et 1) et un réel c tels que :

$$\begin{cases} c \geq \sup_{1 \leq i \leq n_0} T(i) & \\ n \geq n_0 \ : & n(9c/10 + \alpha) \leq ucn \\ & (13c/2 + \beta) \leq (1-u)cn \end{cases}$$

Pour cela il suffit de choisir u et n_0 de manière à obtenir des conditions réalisables pour c, c'est à dire $u - 9/10 > 0$ pour la première équation et $(1-u)n_0 - 13/2 > 0$ pour la deuxième. Choisissons par exemple $u = 19/20$ et $n_0 = 150$. Il suffit alors maintenant que c vérifie les conditions :

$$\begin{cases} c \geq 20\alpha \\ c \geq \beta \\ c \geq \sup_{1 \leq i \leq n_0} T(i) \end{cases}$$