Complexité

235711131723

15 octobre 2022

1 Différents algorithmes de tri

1.1 Tri à bulle

Principe.

- On parcourt le tableau et on permute une case avec la suivante si elles ne sont pas dans le bon ordre
- À la fin du parcours, si on fait une permutation, on recommence le parcours.

Théorème. Le tri à bulle termine correctement en $\Theta(n^2)$.

Algorithme. Entrée : Tableau d'entiers T de taille n.

```
1  Entier temp
2  Bool Permutation := Vrai
3  Tant Que(Permutation):
4      Permutation := Faux
5      Pour(i := 0; i < n - 1; i := i + 1):
6          Si(T[i] > T[i + 1]):
7              temp := T[i]
8              T[i] := T[i + 1]
9              T[i + 1] := temp
10             Permutation := Vrai
11         Fin Si
12     Fin Pour
13 Fin Tant Que
```

Démonstration du théorème. Posons $j \in \mathbb{N}$, pour $0 \leq k \leq n-1$, $H(k)$: "Après k exécution de la boucle **Tant Que**, les j dernières cases du tableau contiennent les j plus grands éléments triés par ordre croissant.".

- **Initialisation de l'invariant.** Quand on arrive à la première exécution de la ligne 3, on a $j = 0$ et l'invariant est trivialement vrai.

— **Sortie de la boucle Tant Que.** On a `Permutation := Faux`. Puisqu'on avait initialement `Permutation := Vrai` alors $j > 0$.

  Puisque `Permutation := Faux` à la fin de la boucle **Tant Que**, alors le test à la ligne 6 est faux et $\forall i \in [\![0, n-2]\!], T[i] < T[i+1]$. Autrement dit, le tableau $T[0 \dots (n-1)]$ est trié dans l'ordre croissant.

  Donc la boucle termine bel et bien correctement.

Tableau déjà entièrement trié. Supposons que $j = n$. Alors l'invariant $H(k)$ devient, pour $0 \leq k \leq n-1$, $H(k)$: "Le tableau est trié par ordre croissant à la k-ième itération.".

— Lors de la première itération de la boucle **Tant Que**, le test de la ligne 6 est toujours faux.
— On sort de cette boucle à la fin de la première itération. Je termine en $O(n)$.
— Je termine.

Invariant de la boucle Pour. Supposons l'invariant $H(k)$ vrai pour $k \in \mathbb{N}$ et supposons que nous sommes à la ligne 4 du code avec `Permutation := Vrai`. Montrons que $H(k+1)$ est vrai.

Posons $M = \max\limits_{i \in [\![0, n-(j+1)]\!]} T[i]$ et m tel que $T[m] = M$. Posons $H'(i)$: "Les dernières cases du tableau sont triés par ordre croissant et $T[\min(n-(j+1), \max(i,m))] = M$.".

Initialisation. Pour $i = 0$ et $H(k)$, les j dernières cases sont triées et contiennent $T[\min(n-(j+1), \max(i,m))] = M$. De plus, $\min(n-(j+1), \max(i,m)) = m$ et $T[m] = M$.

1.2 Tri par insertion

Principe.