UFR IMA

UNIVERSITE JOSEPH FOURIER GRENOBLE 1
SCIENCES, TECHNOLOGIE, MEDECINE

Informatique & Mathématiques Appliquées | Sciences, Technologie, Médecine

L'objectif de cette note de cours est de montrer comment on appréhende un problème d'analyse de coût d'un algorithme. Ici l'analyse peut être menée jusqu'au bout (ou presque) et illustre les différentes méthodes que l'on peut employer.

Soit l'algorithme `tri-rapide` :

fonction `tri-rapide` (ensemble E, entier n)
{E est un ensemble d'objets, l'ensemble des objets est muni d'une relation d'ordre total $\succ$}
si E=$\emptyset$ **alors**
  retourne {}
sinon
  pivot $\leftarrow$ élément(E)
  (E1,E2) = segmentation(E,pivot)
  {E1 est l'ensemble des éléments de E inférieurs ou égaux à pivot}
  {E2 est l'ensemble des éléments de E supérieurs strictement à pivot}
  retourne {tri-rapide(E1),pivot,tri-rapide(E2)}
fin si

L'objectif est d'analyser le coût de cet algorithme en nombre de comparaisons effectuées entre des éléments. Une première étape consiste à se donner des exemples pour comprendre le déroulement de l'algorithme. Ensuite on recherche des exemples présentant des "cas extrêmes" au pire et au mieux. Puis on formalise le problème en écrivant les équations vérifiées par le coût de l'algorithme. On résout ces équations, ou bien, si on ne trouve pas de solution on essaye de majorer/minorer le coût (en ordre de grandeur). Si il y a une grande différence entre le coût au pire et le coût au mieux (par exemple s'il ne sont pas sur la même échelle), on évalue l'ordre de grandeur du coût moyen.

Etape 1 Faite en cours sur un exemple $E = \{4, 3, 6, 7, 2, 5, 1\}$. On choisit de prendre les éléments de E dans l'ordre, le pivot étant le premier élément de l'ensemble.

pivot $\leftarrow$ 4
E1 $\leftarrow$ {3,2,1}
E2 $\leftarrow$ {6,7,5}
$\Rightarrow$ 6 comparaisons
tri-rapide(E1)
  pivot $\leftarrow$ 3
  E1 $\leftarrow$ {2,1}
  E2 $\leftarrow$ { }
  $\Rightarrow$ 2 comparaisons
  tri-rapide(E1)
    pivot $\leftarrow$ 2
    E1 $\leftarrow$ {1}
    E2 $\leftarrow$ { }
    $\Rightarrow$ 1 comparaisons
    tri-rapide(E1)
      pivot $\leftarrow$ 1
      E1 $\leftarrow$ { }
      E2 $\leftarrow$ { }
      $\Rightarrow$ 0 comparaisons
    tri-rapide(E2)
      $\Rightarrow$ 0 comparaisons
  tri-rapide(E2)
    $\Rightarrow$ 0 comparaisons
tri-rapide(E2)
  pivot $\leftarrow$ 6
  E1 $\leftarrow$ {5}
  E2 $\leftarrow$ {7}
  $\Rightarrow$ 2 comparaisons

Total 11 comparaisons. On sent que l'algorithme va être efficace si on arrive à segmenter E en 2 ensembles de même taille.

Etape 2 Exemples extrémaux.

Un premier cas consiste à segmenter au mieux l'ensemble, c'est à dire $|E1| = |E2| = \frac{|E|-1}{2}$, par exemple si E={4,2,1,3,6,5,7} on fera 6+2.2=10 comparaisons. Pour un ensemble de taille $n = 2^k - 1$ notre exemple ferait de l'ordre de $k2^k$ comparaisons (voir plus loin), soit de l'ordre de $n \log n$.

A contrario si la segmentation est très déséquilibrée, par exemple pour E={1,2,3,4,5,6,7} on aura 6+5+4+3+2+1=21 comparaisons. Pour le même exemple de taille n on aurait $1 + 2 + \cdots + n - 1 = \frac{n(n-1)}{2}$.

On peut remarquer que sur 2 exemples on obtient des valeurs de coût ayant des ordres de grandeur différents, ce qui encourage à pousser davantage l'analyse du coût.

Etape 3 Formalisation.

Notons $C(E)$ le coût de l'algorithme en nombre de comparaisons pour l'entrée E de taille n. L'opération de segmentation nécessite $n - 1$ opérations donc le coût de l'algorithme vérifie

$$C(E) = n - 1 + C(E1) + C(E2).$$

avec $C(E) = 0$ si $n = 0$ ou 1. Evidemment ce coût dépend fortement du choix du pivot, c'est à dire de la taille respective de $E1$ et $E2$. Les seules contraintes que l'on connaisse sur $E1$ et $E2$ sont

$$0 \leqslant |E1| \leqslant n - 1;\ 0 \leqslant |E2| \leqslant n - 1;\ |E1| + |E2| = n - 1.$$

Comme d'habitude, notons

$$C_{max}(n) = \max_{|E|=n} C(E) \text{ et } C_{min}(n) = \min_{|E|=n} C(E).$$

Etape 4 Majorants/minorants.

Comme on ne peut pas calculer facilement $C_{max}(n)$, montrons par récurrence que $C_{max}(n) \leqslant n^2$.
En effet, cette assertion est vraie pour $n = 0$, supposons-la vérifiée pour tout k avec $0 \leqslant k \leqslant n-1$ et soit E une instance réalisant le coût maximum. On a

$$C_{max}(n) = C(E) = n - 1 + C(E1) + C(E2) \leqslant n - 1 + \max_{0 \leqslant k \leqslant n-1} \{C_{max}(k) + C_{max}(n-1-k)\}.$$

On applique l'hypothèse de récurrence

$$C_{max}(n) \leqslant n - 1 + \max_{0 \leqslant k \leqslant n-1} k^2 + (n-1-k)^2 = (n-1)^2 \leqslant n - 1 + (n-1)^2 = n(n-1) \leqslant n^2.$$

(la fonction $x^2 + (n-1-x)^2$ est une parabole tournée vers le haut donc maximale sur l'une des extrémité de l'intervalle $[0, n-1]$.) La propriété est donc héréditaire, on en déduit que pour tout n

$$C_{max}(n) \leqslant n^2,$$

comme il existe un exemple dont le coût est $\frac{n(n-1)}{2}$en en conclut que

$$C_{max}(n) = \Theta(n^2)$$

Exercice : Montrer par récurrence que $C_{max}(n) = \frac{n(n-1)}{2}$.
Pour le calcul du coût minimal, on utilise un argument sur la complexité du problème du tri (cf ouvrage de Cormen et al.). Le tri ne peut pas se faire en moins de $\mathcal{O}(n \log n)$ comparaisons. Ici on a une instance du problème qui atteint cette borne (le pivot est la médiane à chaque étape). Donc

$$C_{min}(n) = \Theta(n \log n)$$

Etape 5 Analyse en moyenne.

Pour faire l'analyse en moyenne on suppose que les permutations ont toutes la même probabilité. Cela revient à choisir unifrmément le pivot parmi les n éléments de E.
Ceci nous donne (en utilisant la linéarité de la moyenne)

$$C_{moy}(n) = \mathbb{E}C(E) = n - 1 + \mathbb{E}[C(E1)] + \mathbb{E}[C(E2)];$$

On conditionne par la taille de $E1$ (valeur de la position du pivot dans E moins 1)

$$C_{moy}(n) = n - 1 + \sum_{i=0}^{n-1} (C_{moy}(i) + C_{moy}(n-1-i))\frac{1}{n};$$

en simplifiant la somme

$$C_{moy}(n) = n - 1 + \frac{2}{n}\sum_{i=0}^{n-1} C_{moy}(i).$$

On en déduit, en utilisant la même équation de récurrence au rang $n-1$:

$$\begin{aligned} C_{moy}(n) &= n - 1 + \frac{2}{n}C_{moy}(n-1) + \frac{2}{n}\sum_{i=0}^{n-2}(C_{moy}(i); \\ &= n - 1 + \frac{2}{n}C_{moy}(n-1) + \frac{2}{n}\frac{n-1}{2}\left(C_{moy}(n-1) - (n-2)\right); \\ &= (n-1)\frac{2}{n} + \frac{n+1}{n}C_{moy}(n-1). \end{aligned}$$

En divisant par $n+1$ on déduit

$$\frac{C_{moy}(n)}{n+1} = \frac{4}{n+1} - \frac{2}{n} + \frac{C_{moy}(n-1)}{n} = \sum_{i=1}^{n+1} \frac{2}{i} \simeq 2\log(n+1).$$

D'où

$$C_{moy}(n) = \mathcal{O}(n \log n);$$

ce qui est excellent puisque proche (et au moins sur la même échelle) de la complexité du problème du tri.