Algoritmi 2 (A.A. 2005/2006)
Esercitazione 2

Irene Finocchi [§]

April 21, 2006

1 Mediano di due array ordinati

Siano X ed Y due array ordinati, ciascuno di n elementi. Progettare un algoritmo basato sulla tecnica del divide et impera per trovare il mediano dei $2n$ elementi in tempo $O(\log n)$.

Osserviamo che il mediano di X ed Y è l'elemento che occuperebbe la posizione n se i $2n$ elementi fossero tutti ordinati: avrebbe pertanto n elementi più grandi ed $n-1$ elementi più piccoli. Siano m_x e m_y i mediani di X e Y, rispettivamente: entrambi possono essere ottenuti in tempo $O(1)$, essendo X e Y ordinati. Siano X_1 la metà inferiore di X (escluso m_x), e X_2 la metà superiore di X (escluso m_x). Definiamo analogamente Y_1 e Y_2.

L'algoritmo confronta m_x con m_y ed effettua una opportuna chiamata ricorsiva. Se $m_x < m_y$, restituisce il mediano di:

- X_2 e $Y_1 \cup \{m_y\}$, se n è pari;
- $X_2 \cup \{m_x\}$ e $Y_1 \cup \{m_y\}$, se n è dispari;

Si può dimostrare per contraddizione che il mediano di X e Y non può appartenere alle porzioni di array "scartate" tramite un semplice conteggio di elementi. Infatti:

- Sia n pari. Se il mediano fosse in $X_1 \cup \{m_x\}$, avrebbe $n/2+1+n/2 = n+1$ elementi più grandi. Se il mediano fosse in Y_2, avrebbe $n/2 + n/2 = n$ elementi più piccoli.
- Sia n dispari. Se il mediano fosse in X_1, avrebbe $1+(n-1)/2+1+(n-1)/2 = n+1$ elementi più grandi. Se il mediano fosse in Y_2, avrebbe $1+(n-1)/2+1+(n-1)/2 = n+1$ elementi più piccoli.

[§]Dipartimento di Informatica, Università degli Studi di Roma "La Sapienza", Via Salaria 113, 00198 Rome, Italy. E-mail: {finocchi}@di.uniroma1.it.

Il caso $m_x > m_y$ è completamente simmetrico. Se infine $m_x = m_y$, allora quello è il mediano cercato e l'algoritmo può terminare.

Ad ogni passo, la dimensione degli array su cui si effettua la chiamata ricorsiva viene (approssimativamente) dimezzata spendendo tempo costante, da cui la relazione di ricorrenza che descrive il tempo di esecuzione dell'algoritmo è $T(n) = T(n/2) + O(1)$. La soluzione è $T(n) = \Theta(\log n)$, come segue facilmente dal Teorema Master.

2 Punto fisso

Data una sequenza ordinata di n interi distinti, sia positivi che negativi, $a_1 < a_2 < \ldots < a_n$, un punto fisso è un indice i tale che $a_i = i$. Progettare un algoritmo che risolve il problema della ricerca di un punto fisso in tempo $O(\log n)$.

È sufficiente simulare la ricerca binaria, distinguendo i seguenti tre casi:

- Se $a_{n/2} = n/2$ restituisci `true`
- Se $a_{n/2} < n/2$ prosegui a destra, ovvero sugli elementi $a_{n/2+1} \ldots a_n$. Poiché gli elementi sono tutti distinti, $a_{n/2-1} \leq a_{n/2} - 1 < n/2 - 1$. Proseguendo in questo modo, si può dimostrare facilmente che $a_i < i$ per ogni $i < n/2$.
- Se $a_{n/2} > n/2$ prosegui a sinistra, ovvero sugli elementi $a_1 \ldots a_{n/2-1}$. Poiché gli elementi sono tutti distinti, $a_{n/2+1} \geq a_{n/2} + 1 > n/2 + 1$. Proseguendo in questo modo, si può dimostrare facilmente che $a_i > i$ per ogni $i > n/2$.

3 Moltiplicazione di interi di grandezza arbitraria

L'esercizio è svolto nel paragrafo 10.1 del libro "Algoritmi e Strutture Dati", autori: Demetrescu, Finocchi e Italiano, casa editrice: McGraw-Hill.