

# Algoritmica per Informatica Umanistica

## Secondo Appello: 4/2/2011

### Soluzione

#### Esercizio 1

Sia  $A$  un array che contiene  $n$  elementi distinti. Si dia il codice di un algoritmo che modificando il Selection-Sort, produca come risultato il  $k$ -esimo elemento nell'ordinamento. Si determini la complessità dell'algoritmo.

```
Selectionk (A, n):  
for (i=0; i<k; i++){  
    min=A[i]; indicemin=i;  
    for (j=i+1; j<n; j++){  
        if A[j]< min {  
            min = A[j];  
            indicemin = j;  
        }  
    }  
    A[indicemin]= A[i]; A[i]= min;  
}  
return A[k-1];
```

La complessità di Selectionk è  $O(nk)$

#### Esercizio 2

Confrontare dal punto di vista della complessità di tempo, la soluzione data dell'esercizio 1 con un'altra possibile, che usa MERGESORT per risolvere lo stesso problema.

Per risolvere il problema dell'esercizio 1 si può in alternativa ordinare tutto l'array e poi dare come risultato l'elemento in posizione  $k-1$ , cioè  $A[k-1]$ . Questa soluzione ha costo  $O(n \log n) + O(1)$  cioè  $O(n \log n)$  in quanto Mergesort costa  $O(n \log n)$  e l'accesso all'elemento in posizione  $k$  ha costo costante. Dunque se  $k$  è piccolo e più precisamente  $< O(\log n)$  la soluzione dell'esercizio 1 sarà più vantaggiosa, altrimenti conviene l'altra.

#### Esercizio 3

Scrivere il codice di un algoritmo ricorsivo qualsiasi che abbia associata la seguente relazione di ricorrenza:

$$\begin{aligned} T(n) &= 0 && \text{per } n=1 \\ T(n) &= 2T(n/2) + 1 && \text{per } n>1. \end{aligned}$$

L'algoritmo seguente ha la complessità voluta e calcola la somma degli elementi dell'array  $A$  con la tecnica del divide et impera.

```
Somma (A, sx, dx) :  
if sx == dx return A[sx];  
else {cx = (sx+dx)/2;  
    return (Somma (A, sx, cx) + Somma (A, cx+1, dx));  
}
```

#### Esercizio 4

Si indichi la matrice di programmazione dinamica relativa all'EDIT DISTANCE tra le due sequenze  $X=ABABABB$  e  $Y=BABBABA$  e quella relativa alle APPARIZIONI APPROSSIMATE della sequenza  $Y=BAA$  nella sequenza  $X$ .\*

	0	A	B	A	B	A	B	B	Edit Distance
0	0	1	2	3	4	5	6	7	
B	1	1	1	2	3	4	5	6	
A	2	1	2	1	2	3	4	5	
B	3	2	1	2	1	2	3	4	
B	4	3	2	3	2	3	3	3	
A	5	4	3	3	3	2	3	4	
B	6	5	4	4	3	3	2	3	
A	7	6	5	4	4	3	3	3	

	0	A	B	A	B	A	B	B	Apparizioni approssimate
0	0	0	0	0	0	0	0	0	
B	1	1	0	1	0	1	0	0	
A	2	1	1	0	1	0	1	1	
B	3	2	2	1	1	1	1	2	

#### Esercizio 5

Indicare il numero **medio** di confronti fatti dall'algorithm di Ricerca Sequenziale.

L'algorithm di ricerca sequenziale esegue un passo (1 confronto) se l'elemento cercato si trova in prima posizione, due passi se si trova in seconda ecc. Se l'elemento cercato si trova in ultima posizione fa  $n$  passi e se non appartiene all'array ne fa ancora  $n$ .

Per trovare il numero di passi in media si fa la somma e si divide per  $n$ .

$$T(n) = (1+2+ \dots +n +n)/n = ((n+1)n + n)/2n = (n+2)/2$$