

## Algoritmi e Strutture Dati – 16/06/14

**Esercizio 0** Scrivere correttamente nome, cognome, numero di matricola, riga e colonna.

### Esercizio 1 – Punti $\geq 6$ (Parte A)

Trovare un limite superiore, il più stretto possibile, per la seguente equazione di ricorrenza, utilizzando il metodo di sostituzione.

$$T(n) = \begin{cases} 2T(\lfloor n/8 \rfloor) + \sqrt[3]{n} & n > 1 \\ 1 & n \leq 1 \end{cases}$$

### Esercizio 2 – Punti $\geq 6$ (Parte A)

Scrivere un algoritmo che, dato un grafo non orientato  $G$ , conti il numero delle sue componenti connesse che sono anche alberi.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

### Esercizio 3 – Punti $\geq 6 + BONUS$ (Parte B)

Siano dati in input  $n$  file; il file  $i$ -esimo che ha lunghezza  $L[i]$  e probabilità di accesso  $P[i]$  tale che  $\sum_{i=1}^n P[i] = 1$ . Questi file devono essere registrati su un dispositivo a nastro, in cui per leggere il file  $i$ -esimo è sempre necessario far scorrere il nastro dall'inizio fino al termine del file stesso, leggendo tutti i file memorizzati prima di esso. Il vostro compito è trovare un ordinamento dei file sul nastro, in modo da minimizzare il tempo atteso totale di accesso. Dato un possibile ordinamento  $F_1 \dots F_N$ , definiamo:

- Il *tempo di accesso*  $T[i]$  del file  $i$ -esimo è dato dalla lunghezza del file e dalla somma delle lunghezze di tutti i file che lo precedono.

$$T[i] = \sum_{k=1}^i L[k]$$

- Il *tempo atteso di accesso*  $E[i]$  del file  $i$ -esimo è dato dal suo tempo di accesso moltiplicato dalla sua probabilità di accesso.

$$E[i] = T[i] \cdot P[i]$$

- Il *tempo atteso totale di accesso* è la somma dei tempi attesi di accesso di tutti i file.

$$Tot = \sum_{i=1}^n E[i]$$

Problemi:

1. Si consideri il seguente ordinamento greedy, e si dimostri (con un controesempio) che non risolve il problema: si ordinino i file per probabilità di accesso  $P[i]$  decrescente.
2. Si consideri il seguente ordinamento greedy, e si dimostri (con un controesempio) che non risolve il problema: si ordinino i file per dimensione  $L[i]$  crescente.
3. BONUS: (i) Si individui un ordinamento greedy che risolva il problema e (ii) (MOLTO DIFFICILE) si dimostri che tale ordinamento risolve il problema.

### Esercizio 4 – Punti $\geq 12$ (Parte B)

Siete alla guida di un'auto elettrica che ha  $A$  km di autonomia. Percorrete una strada caratterizzata da  $n$  distributori di elettricità. Il prezzo dell'elettricità è uguale in tutti i distributori, mentre cambia il tempo di rifornimento a seconda del numero e del tipo di caricabatteria. Al distributore  $i$ -esimo, il tempo per rifornirsi di energia sufficiente per percorrere ulteriori  $g$  chilometri è pari a  $g \cdot t[i]$ . Il distributore  $i$ -esimo si trova a  $d[i] \leq A$  chilometri dal precedente. Iniziate il percorso al distributore 1 con il serbatoio vuoto. Dovete arrivare al distributore  $n$ , anche con autonomia 0.

Scrivete un algoritmo che preso in input il vettore dei tempi  $t[]$ , il vettore delle distanze  $d[]$ , l'autonomia  $A$  e il numero di distributori  $n$ , restituisca la minor quantità di tempo speso in rifornimenti che è necessaria per percorrere la strada.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.