Introduzione agli Algoritmi Esame Scritto a canali unificati

docenti: T. Calamoneri, A. Monti Sapienza Università di Roma 31 Gennaio 2021

Esercizio 1 (8 punti):

Si consideri la seguente funzione:

```
funzione \operatorname{Exam}(n): tot \leftarrow n; if n <= 1: return tot; j \leftarrow 512; while j >= 2 do: k \leftarrow 0; while 3*k <= n do: k \leftarrow k+1; tot \leftarrow tot + Exam(k); j \leftarrow j/2; return tot
```

- a) Si imposti la relazione di ricorrenza che ne definisce il tempo di esecuzione giustificando dettagliatamente l'equazione ottenuta.
- b) Si risolva la ricorrenza usando il **metodo principale** (o un altro metodo, ricordando che $\sum_{i=1}^{k} 3^{i} = \Theta(3^{k})$) dettagliando i passaggi del calcolo e giustificando ogni affermazione.

NOTA: Se necessario, usare le seguenti convenzioni:

```
- anziché \leq o \geq scrivere <= o >= - anziché \Theta e \Omega scrivere Teta e Omega - anziché \sum_{i=0}^k scrivere S[i=0, k] - anziché a^b scrivere a**b.
```

Esercizio 2 (12 punti):

Sia A un array di dimensione n e B un array **ordinato** di dimensione m, contenenti entrambi numeri interi. Si vuole trovare il numero di interi di A che non sono presenti in B. Progettare un algoritmo ricorsivo che risolva il problema con un costo computazionale asintotico strettamente inferiore a $\Theta(nm)$.

Ad esempio: per A = [8, 1, 2, 12, 10, 11, 20, 2] e B = [3, 3, 4, 8, 10, 10, 13, 20, 21, 22] l'algoritmo deve restituire 5 (i numeri in A e non in B sono infatti 1, 2, 2, 11, 12.

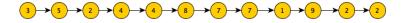
Dell'algoritmo proposto

- a) si dia la descrizione a parole,
- b) si scriva lo pseudocodice,
- c) si giustifichi il costo computazionale.

Esercizio 3 (10 punti):

Si consideri una lista L, in cui ogni elemento è un record a due campi, il campo val contenente un intero ed il campo next con il puntatore al nodo seguente (next vale None per l'ultimo record della lista.

Bisogna contare i record della lista contenenti numeri pari. Si consideri ad esempio la lista L, per questa lista bisogna la risposta è 6



Progettare un algoritmo ricorsivo che, dato il puntatore r alla testa della lista effettui l'operazione di conteggio in tempo $\Theta(n)$ dove n è il numero di elementi presenti nella lista.

Dell'algoritmo proposto

- a) si dia la descrizione a parole,
- b) si scriva lo pseudocodice,
- c) si giustifichi il costo computazionale risolvendo la ricorsione che viene fuori dall'algoritmo utilizzando uno dei metodi di soluzione visti a lezione.