

## Problem Set 2

docente: Luciano Gualà

### Esercizio 1 (equazioni di ricorrenza)

Si risolvano le seguenti equazioni di ricorrenza. Si assuma sempre  $T(\text{costante}) = O(1)$ .

(a)  $T(n) = T(n - 10) + 10$ .

(b)  $T(n) = T(n/2) + 2^n$ .

(c)  $T(n) = T(n/3) + T(n/6) + n^{\sqrt{\log n}}$ .

(d)  $T(n) = T(\sqrt{n}) + \Theta(\log \log n)$ .

(e)  $T(n) = T(n/2 + \sqrt{n}) + \Theta(1)$ .

(f)  $T(n) = \sqrt{n}T(\sqrt{n}) + n$ .

### Esercizio 2 (visite di alberi)

Sia  $T$  un albero binario di  $n$  nodi con radice  $r$  in cui ad ogni nodo  $v$  è associato un valore positivo. Dato un cammino  $P$  radice-foglia, diremo che  $P$  è *crescente* se i valori dei nodi di  $P$  costituiscono una sequenza crescente quando si guardano i nodi dalla radice verso la foglia. Si assuma che  $T$  è mantenuto attraverso una struttura dati collegata e che ogni nodo  $v$  è rappresentato con un record che contiene come campi i puntatori al padre e ai figli ( $v.p$ ,  $v.s$ ,  $v.d$ ) e il valore del nodo ( $v.val$ ). Si progetti un algoritmo con complessità  $O(n)$  che, preso  $T$ , restituisca il numero di cammini radice-foglia crescenti contenuti in  $T$ . Si fornisca lo pseudocodice dettagliato dell'algoritmo e possibilmente non si usino variabili globali e passaggi di parametri per riferimento.

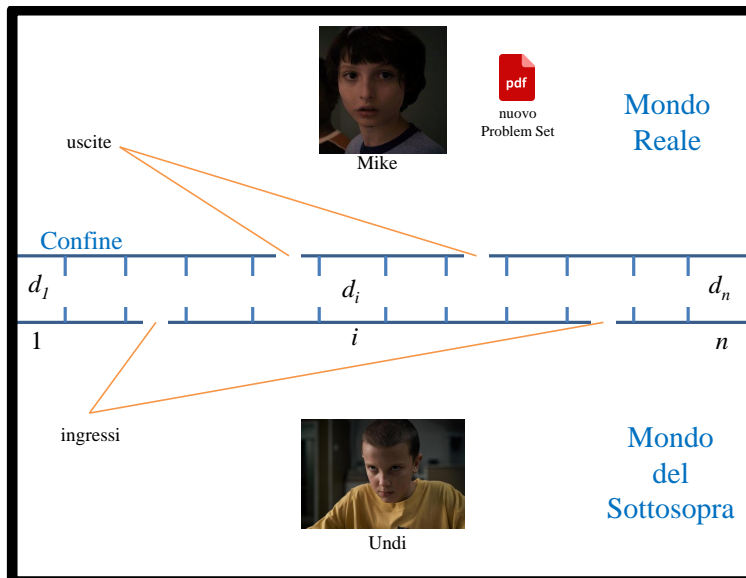
### Esercizio 3 (strana cosa, l'amicizia)

Undi, una testarda ragazzina con poteri telecinetici, vuole raggiungere il suo amico nerd Mike per poter finalmente passare il pomeriggio insieme e lavorare sul nuovo Problem Set di Algoritmi e strutture dati. Purtroppo però Undi è bloccata nel *Mondo del Sottosopra* e per raggiungere Mike nel *Mondo Reale* deve attraversare il *Confine*. Il Confine è una striscia di realtà sospesa fra i due mondi costituita da  $n$  stanze comunicanti. Alcune di queste stanze hanno un *ingresso* (una porta che Undi può usare per passare dal Mondo del Sottosopra al Confine) mentre altre hanno un'*uscita* (una porta che Undi può usare per uscire dal Confine ed entrare nel Mondo Reale). Le stanze del Confine sono presidiate da *demodog*, esseri malvagi a metà fra cani e demoni. In particolare, Undi conosce, per ogni stanza  $i$ , il numero  $d_i$  di demodog che sono nella stanza  $i$ . Così come conosce le posizioni degli ingressi e delle uscite. Un'illustrazione della situazione è fornita in Figura 1.

Quello che Undi può fare è entrare nel Confine usando un ingresso di una stanza  $i$ , attraversare un segmento di Confine fino ad arrivare ad una stanza  $j$  in cui c'è un'*uscita* e raggiungere finalmente Mike (e il nuovo Problem Set). Nel fare il percorso dalla stanza  $i$  alla stanza  $j$  deve però affrontare tutti i demodog che incontra, sperando che non siano troppi.

Vi chiedo di aiutare Undi progettando un algoritmo con complessità temporale  $O(n)$  che sia in grado di calcolare un percorso che consenta ad Undi di raggiungere Mike affrontando il minor numero di demodog possibile.

Figura 1: Il Confine che separa Undi e Mike (e il nuovo Problem Set).



**Esercizio 4** (*Strutture dati: mantenere dinamicamente il mediano*)

Si progetti una struttura dati che mantiene un insieme  $S$  di elementi con chiavi prese da un dominio totalmente ordinato soggetto alle seguenti operazioni:

- **Insert**( $S, k$ ): inserisce un nuovo elemento di valore  $k$  in  $S$ .
- **Delete**( $S, x$ ): rimuove l'elemento  $x$  da  $S$ . Si pensi ad  $x$  come al puntatore (riferimento diretto) all'elemento da cancellare.
- **Median**( $S$ ): restituisce (senza rimuoverlo) il valore dell'elemento mediano contenuto in  $S$ ; ovvero, se indichiamo con  $n$  il numero di elementi contenuti in  $S$ , si vuole restituire il valore dell' $\lceil \frac{n}{2} \rceil$ -esimo minimo di  $S$ .

Tutte le operazioni devono avere complessità temporale (asintoticamente) logaritmica nel numero di elementi presenti nella struttura dati.