

## Traccia prova d'esame

I fabbri della Goron City hanno deciso di fare un piano strategico per l'anno 2020 riguardante l'approvvigionamento di nuovi materiali. I materiali di cui hanno interesse sono usati in alcuni processi di costruzione. Ogni processo P è formato da un insieme di istruzioni I, etichettate con numeri interi diversi tra loro, ed è strutturato come un albero binario, la cui radice è un'istruzione speciale etichettata con valore 0. Dati n processi in input, ci viene chiesto di restituire in output la coppia  $(d_{min}, d_{max})$ , dove  $d_{min}$  e  $d_{max}$  sono rispettivamente la profondità minima e massima tra tutti i processi in P. La profondità di un processo è il massimo livello delle sue istruzioni; se un processo ha solo una istruzione, la profondità è uguale ad 1.

### Struttura dell'input

L'input consiste in una prima riga nel formato n dove n è il numero di processi dati in input. Ogni processo consiste in una serie di righe terminate da -1. La prima riga di un processo contiene sempre 0, indicante la radice del processo. Le righe successive sono nel formato a:b x dove a è un numero indicante una nuova istruzione, b è un numero che indica una istruzione già presente nel processo e x può essere uno dei due simboli s oppure d. Il simbolo s indica che l'istruzione a deve essere inserita come figlio sinistro dell'istruzione a, altrimenti come figlio destro (nel caso del simbolo a).

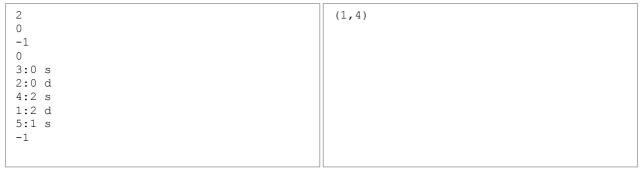
#### Struttura dell'output

L'output consiste nella coppia  $(d_{min},\ d_{max})$ , dove  $d_{min}$  e  $d_{max}$  sono rispettivamente la profondità minima e massima tra tutti i processi dati in input.

### Esempi input – output



Il primo processo ha profondità 3, il secondo 3 e il terzo 2. La coppia  $(d_{min}, d_{max})$  è data quindi da (2,3).



Il primo processo ha profondità 1, il secondo 4. La coppia  $(d_{min},\,d_{max})$  è data quindi da (1,4).

# Algoritmi e Strutture Dati Prova d'esame – 27/02/2020



#### Regole e istruzioni

- Si può scegliere se usare C++ o Java; in entrambi i casi, si presuppone che lo studente sappia compilare il codice sorgente e avviare l'eseguibile ottenuto tramite terminale/prompt dei comandi.
- Si può assumere che l'input sia sempre corretto.
- Il vostro programma deve **leggere da stdin** e **scrivere su stdout**. La lettura da file o l'hardcoding di un input nel vostro programma **non rappresenta una soluzione corretta**.
- Per effettuare le (vostre) varie prove, potete creare dei file testuali contenenti input di prova e
  - O Scrivere o copiare riga per riga il vostro input al programma,
  - o Reindirizzare il contenuto del file di input al vostro programma (consigliato).
- Il vostro programma verrà valutato su vari input utilizzando il secondo metodo.

### Come posso reindirizzare su stdin?

Supponendo di utilizzare C++ e di essere su Linux/OS X

```
cat input.txt | ./programma
```

dove input.txt è un file testuale contenente un input e programma è l'eseguibile ottenuto dalla compilazione del vostro codice sorgente (input.txt e programma devono essere nella stessa cartella)

Supponendo di utilizzare C++ e di essere su Windows

```
type input.txt | programma.exe
```

dove input.txt è un file contenete un input e programma.exe è l'eseguibile ottenuto dalla compilazione del vostro codice sorgente (input.txt e programma.exe devono essere nella stessa cartella).