

Laboratorio

12 Aprile 2012

Qsort e struct

Scrivere un programma che utilizzi la procedura *qsort* per ordinare un vettore di punti del piano cartesiano. Ciascun punto è formato da una coppia di coordinate (x, y) . I punti devono essere ordinati per ascissa crescente. A parità di ascissa, si ordina per ordinata decrescente.

Il formato dell'input è il seguente. Il primo numero è la lunghezza N dell'array seguito dagli N coordinate, una per riga. Per ciascun punto si riportano ascissa e ordinata separate da uno spazio.

Il programma deve stampare in output la sequenza ordinata con un punto per riga.

Input	Output
5	2 7
2 5	2 5
12 2	2 2
2 7	3 4
3 4	12 2
2 2	

Qsort: Stringhe e struct

Scrivere un programma che utilizzi la procedura *qsort* per ordinare un array di stringhe. Le stringhe devono essere ordinate per lunghezza. A parità di lunghezza, si utilizza l'ordinamento lessicografico. **Utilizzare una struct che memorizzi una stringa e la sua lunghezza per evitare di calcolare quest'ultima ad ogni confronto.**

Il formato dell'input è il seguente. Il primo numero è la lunghezza N dell'array seguito dagli N stringhe, una per riga.

Il programma deve stampare in output la sequenza ordinata con una stringa per riga.

Input

```
5
pizza
piazza
piano
ciao
tre
```

Output

```
tre
ciao
piano
pizza
piazza
```

Punti e colori

Prova di esame del 14/07/2009.

Si consideri il quadrante positivo del piano cartesiano. Un punto colorato sul piano è caratterizzato da una tripla (x, y, c) , dove x , y e c sono valori interi non-negativi. Il primo intero della tripla caratterizza l'ascissa del punto, il secondo intero l'ordinata, il terzo intero è il colore assegnato al punto. Sia A un insieme di N punti colorati. Lo scopo del programma è quello di rispondere a una sequenza di interrogazioni sui punti di A . Un'interrogazione è definita da due coppie (x_1, y_1) e (x_2, y_2) , dove $x_1 < x_2$ e $y_1 < y_2$, che identificano un rettangolo R :

$$R = \{(u, v) \in \mathbb{N}^2 \mid x_1 \leq u \leq x_2, y_1 \leq v \leq y_2\}$$

Data un'interrogazione R si vuole calcolare il numero di colori distinti dei punti di A che ricadono in R (i punti sul perimetro del rettangolo devono essere considerati nel conteggio).

Scrivere un programma che legga da tastiera una sequenza A di N punti colorati e un insieme Q di M interrogazioni, e stampi la risposta a ciascuna interrogazione su una riga distinta. Nel caso non vi siano punti all'interno del rettangolo stampare 0.

L'input è formattato nel seguente modo: le prime due righe contengono i due interi N e M , rispettivamente. Si assuma che $N > 0$ ed $M > 0$. Seguono $N + M$ righe. Le prime N righe contengono i punti colorati, uno per riga. Ogni punto è definito da 3 interi, separati da uno spazio, che rappresentano, nell'ordine, i valori x , y e c . Le ultime M righe contengono le interrogazioni, disposte una per riga. Ogni interrogazione è definita da 4 interi, separati da uno spazio, che rappresentano, nell'ordine, i valori x_1 , y_1 , x_2 e y_2 . Si assuma che $x_1 < x_2$ e $y_1 < y_2$.

L'output deve contenere solo e soltanto gli interi di risposta alle interrogazioni.

NOTA: A parte essere contenute in un intero int del C, non si possono fare ulteriori assunzioni sulla grandezza delle coordinate e dei colori.

Input

```
6 2 7 8
0 2 5 5
6 4
0 0 1
6 0 1
6 1 13
1 3 8
4 4 9
4 6 137000
2 2 9 7
0 0 7 7
```

Output

2
5
0
2

Puzzled

Elemento maggioritario

Un array A contiene n interi. Uno di essi è detto *elemento maggioritario* se occorre in A almeno $\lfloor \frac{n}{2} \rfloor + 1$ volte. Si vuole un algoritmo che identifichi l'elemento maggioritario, se presente, in tempo $O(n)$ utilizzando $O(1)$ spazio aggiuntivo. L'algoritmo deve stampare N se non è presente alcun elemento maggioritario.

Caso particolare. Si assuma che tutti gli elementi, ad esclusione eventualmente del maggioritario, occorranza una sola volta in A .

Caso generale. Gli elementi possono avere un numero arbitrario di occorrenze.

Input	Output
5 1 10 22 11 2	N
Input	Output
5 22 10 22 11 22	22

Puzzled
Due uova
Interview di Google

In un palazzo di 100 piani si vuole stabilire qual è il piano più alto dal quale è possibile lanciare un uovo senza che esso si rompa. Le uova sono considerate tutte aventi lo stesso grado di resistenza.

Si hanno a disposizione solamente due uova e si vuole individuare tale piano con il minor numero di lanci.

Soluzione banale: si provano tutti i piani sequenzialmente dal primo al novantanovesimo. Dopo ogni lancio si prosegue se e soltanto se l'uovo appena lanciato risulta ancora intatto. Al caso pessimo sono necessari 99 lanci.