

Corso di Laurea Magistrale in Ingegneria Informatica

CORSO DI ALGORITMI E STRUTTURE DATI

Prof. ROBERTO PIETRANTUONO

Prima prova in itinere – primo turno

Indicazioni

La consegna prevede un file “CognomeNome.PDF” in cui è riportato l’implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell’algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore $O(f(n))$). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, “la complessità è $O(n \log n)$ al netto della complessità dell’algoritmo x , che è non nota”). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

PROBLEMA 1

Il signor Rossi si trasferisce a Napoli. Ha una famiglia molto numerosa a Napoli, e tutti vivono a via Roma. Dal momento che intende visitare spesso tutti i suoi parenti, sta cercando di trovare casa vicino a loro. Il signor Rossi vuole ridurre al minimo la distanza totale da tutti loro (ossia, vuole determinare la posizione ottimale della casa che minimizza la somma delle distanze dalle case dei suoi parenti), e ti ha ingaggiato per scrivere un programma che risolva il problema.

INPUT

L'input è costituito da diversi casi di test. La prima riga contiene il numero di casi di test. Per ogni test case è fornito un numero intero che indica il numero dei parenti r ($0 < r < 500$), ed i numeri civici (anche essi interi) $s_1, s_2, \dots, s_i, \dots, s_r$ dove vivono ($0 < s_i < 30000$). Nota che diversi parenti potrebbero vivere allo stesso civico.

OUTPUT

Per ogni test case, il programma riporti in output la *somma minima* delle distanze dalla casa del signor Rossi alle case di ciascuno dei suoi parenti. La distanza tra due numeri civici s_i ed s_j è $d_{ij} = |s_i - s_j|$.

Sample Input

```
2
2 2 4
3 2 4 6
```

Sample Output

```
2
4
```

PROBLEMA 2

Un artigiano deve svolgere N lavori (ordini dai clienti). Egli può svolgere un solo lavoro al giorno. Per ogni i -esimo lavoro, è noto T_i ($1 \leq T_i \leq 1000$), un intero che indica il tempo in giorni impiegato dall'artigiano per finire il lavoro. Per ogni giorno di ritardo accumulato prima di iniziare a lavorare l' i -esimo ordine, l'artigiano deve pagare una penale di S_i ($1 \leq S_i \leq 10000$) centesimi di euro. Si implementi un programma per trovare la sequenza dei lavori che comporta la minima somma di penali.

INPUT

L'input inizia con un singolo intero positivo su una riga a sé stante, indicante il numero dei casi di test successivi, seguita da una riga vuota. Le righe successive descrivono il caso di test; la prima di tali righe contiene un numero intero N ($1 \leq N \leq 1000$), le successive N righe contengono ciascuna due numeri: il tempo impiegato e la penale di ogni attività (la prima coppia di numeri si riferisce al lavoro $i=1$, seconda coppia il lavoro $i=2$, ecc.).

OUTPUT

Per ogni caso di test, il programma dovrà stampare la sequenza dei lavori che comporta una penale minima. Ogni lavoro è identificato dall'ordine in cui è apparso in input, i . Tutti i numeri interi devono essere posizionati su una sola riga di output e separati da uno spazio. Se sono possibili più soluzioni, stampare la prima soluzione in ordine lessicografico.

Osservazione: data una coppia di lavori i, j , la decisione di schedulare prima i o prima j non impatta sul costo totale dei restanti $N-2$ lavori; non è dunque necessario controllare ogni possibile permutazione.

Sample Input

```
2

4
3 4
1 1000
2 2
5 5

3
123 123
3 3
5 5
```

Sample Output

```
2 1 3 4
1 2 3
```