DIETI UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

CORSO DI ALGORITMI E STRUTTURE DATI

Prof. ROBERTO PIETRANTUONO

Indicazioni

Si consegni la prova cartacea indicando Nome Cognome e Matricola. Tempo: 1h e 30'.
Si riporti la soluzione al problema richiesto in pseudocodice, secondo le convenzioni descritte al corso; opzionalmente, se si ritiene necessario, si può commentare lo pseudocodice (o riga per riga o alla fine). Va infine riportata la complessità temporale dell'algoritmo (complessità nel caso peggiore, è sufficiente il limite superiore O(f(n))). Non occorre una dimostrazione, ma una semplice giustificazione per la complessità riportata. Se si utilizzano funzioni di libreria (ad esempio, *sort*, *binary_search*, ...) di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, "la complessità è O(log n) al netto della complessità della funzione di libreria *x*, che è non nota").

PROBLEMA 1

Dato un array non ordinato, si implementi un algoritmo con approccio DIVIDE ET IMPERA per trovare almeno un massimo locale.
Un elemento è un massimo locale se il suo valore è maggiore o uguale ai valori degli elementi adiacenti (se esistono). Possono esserci più massimi locali, nel qual caso è richiesto che l'algoritmo ne restituisca uno qualsiasi e termini.
Dalla definizione di massimo locale si evince che:

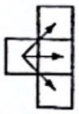
- Se l'array è ordinato in senso crescente, l'ultimo elemento è sempre un massimo locale.
- Se l'array è ordinato in senso decrescente, il primo elemento è sempre un elemento di picco.
- Se tutti gli elementi sono uguali, ogni elemento è un elemento di picco.

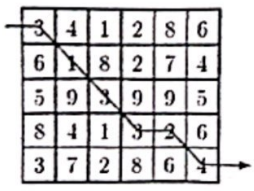
Sample Input
Array = [3, 5, 9, 3]
Array = [3, 5, 9, 11, 14]
Array = [3, 3]
Array = [3, 5, 4, 7, 6]

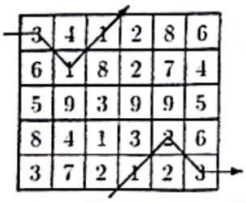
Sample Output
Output = 9
Output = 14
Output = 3
Output = 5 oppure 7

PROBLEMA 2

Data una matrice di *m*×*n* pesi interi positivi, scrivere un programma che calcoli un percorso di peso minimo. Un percorso inizia ad una qualsiasi riga nella prima colonna e consiste in una sequenza di passi che termina nell'ultima colonna. Un passo consiste nel viaggiare dalla colonna *i* alla colonna *i + 1* sulla stessa riga o su una riga adiacente (in diagonale). La prima e l'ultima riga (righe 1 e *m*) di una matrice sono considerate adiacenti. I passi consentiti sono illustrati a destra. Il peso totale di un percorso è la somma dei numeri interi in ciascuna delle *n* celle della matrice che vengono visitate. Ad esempio, di seguito sono mostrate due matrici 5×6.







L'algoritmo stampi il percorso di peso minimo ed il relativo costo. Il percorso è stampato come una sequenza di interi (separati da spazio) che rappresentano le righe attraversate. Se c'è più di un percorso minimo, se ne stampi uno qualsiasi.

Sample Input 1
56
341286
618274
593995
841326
372864

Sample Input 2
56
341286
618274
593995
841326
372123

Sample Output
123445
16
121545
11