



Esempio di compito su complessità computazionale e algoritmi di ordinamento

1. [2 punti] Descrivi qual è lo scopo della notazione O grande e danne una definizione precisa in senso matematico, spiegandone il significato con un esempio.
2. [2 punti] Descrivi a parole ed eventualmente attraverso l'utilizzo di disegni, il funzionamento dell'algoritmo Insertion Sort non adattativo (senza scrivere codice). Sempre a parole descrivi quali sono le idee che permettono di migliorarne le prestazioni, rendendolo adattativo, descrivendo eventuali problemi che possono sorgere e come devono essere gestiti.
3. [2 punti] Scrivi il codice di un algoritmo di ordinamento a tua scelta fra quelli appartenenti alla classe di complessità computazionale quadratica. L'algoritmo dovrà essere implementato all'interno di una funzione che avrà questo prototipo: **funzione(int a[], int l, int r)**, dove a è il vettore di interi da ordinare, l è l'indice dell'elemento inferiore del vettore e r di quello superiore. Al posto di *funzione* andrà il nome dell'algoritmo scelto.
4. [2 punti] Scrivi tutto il codice della funzione di merging dell'algoritmo mergesort, che abbia il seguente prototipo **int mege(int a[],int l, int m, int r)**, dove a è il vettore di interi su cui fare il merging, l è l'indice dell'elemento inferiore del vettore, m l'indice di fine del primo sottovettore ordinato e r di quello superiore.
5. [Difficile 1 punto] Dato un vettore contenente N interi **non ordinati**, trovare un algoritmo che permetta di comunicare l' n -esimo elemento in un tempo proporzionale a $O(N)$ nel caso medio e scriverne il codice. L' n -esimo elemento è l'elemento in posizione n all'interno del vettore ordinato. Es. se il vettore contenesse i numeri 3 45 23 34 12 78 9 il terzo elemento sarebbe 12, il quinto sarebbe 34 ecc.