

INFORMATICA GENERALE - I-Z

Claudia Malvenuto - Daniele A. Gewurz
Scheda esercizi n. 5

1. Dimostrare, per induzione sull'altezza dell'albero, che in uno heap il massimo è nella radice.
2. Scrivere un algoritmo in pseudocodice che implementa la funzione di inserimento di un elemento in uno heap rappresentato con un vettore (cioè la funzione che, dato un heap e un elemento, restituisce un nuovo heap con tutti gli elementi di quello di partenza più quello nuovo inserito correttamente) e valutarne la complessità. Valutare la complessità di un algoritmo che crea un heap partendo da un heap vuoto e da un vettore qualsiasi di interi, e inserendoli uno alla volta.
3. Scrivere la funzione "AggiustaHeap" (chiamata anche "FixHeap", "MaxHeapify" etc.) per un albero rappresentato con un vettore (cioè la funzione che, dato un vettore A e un indice i , e supposto che i due sottoalberi con radici nei figli di $A[i]$ siano heap, fa scendere l'elemento $A[i]$ in modo che l'albero radicato nell' i -esimo elemento sia un heap).
4. (a) Scrivere in versione iterativa la procedura "CreaHeap" per un vettore, usando ripetutamente "AggiustaHeap" (scrivere cioè una funzione che, dato un vettore qualsiasi, lo "aggiusti" finché non diventa un heap).
(b) Dimostrare con un invariante di ciclo la correttezza di "CreaHeap" (iterativa).
5. Dimostrare con un controesempio che l'algoritmo heapsort non è stabile.
6. Dato il seguente vettore, applicare (a mano) la procedura "CreaHeap" per farlo diventare un heap: (35, 40, 20, 15, 7, 3, 1, 9, 11, 6).
7. Scrivere e studiare la complessità di una versione sbilanciata del mergesort in cui, delle due chiamate ricorsive, una viene effettuata sul primo quarto del vettore da ordinare e l'altra sui restanti tre quarti.
8. In un vettore V di n interi chiamiamo *scalino* un indice i ($1 \leq i < n$) tale che $V[i] < V[i + 1]$.
(a) Dato un vettore V di $n \geq 2$ interi tale che $V[1] < V[n]$, dimostrare che V ha almeno uno scalino.
(b) Progettare un algoritmo che, dato un vettore V di $n \geq 2$ interi tale che $V[1] < V[n]$, trovi uno scalino in un tempo $O(\log n)$.
9. Un vettore di interi V è detto *unimodale* se ha tutti valori distinti ed esiste un indice i tale che $V[1] > V[2] > \dots > V[i - 1] > V[i]$ e $V[i] < V[i + 1] < V[i + 2] < \dots < V[n]$, dove n è la dimensione del vettore. Progettare un algoritmo che, dato un vettore unimodale, restituisca il valore minimo del vettore. La complessità dell'algoritmo deve essere $O(\log n)$.
10. Progettare un algoritmo che, dato un vettore V di n interi, calcoli il numero di inversioni. Un'inversione è una coppia di indici i, j tali che $i < j$ e $V[i] > V[j]$. L'algoritmo deve avere complessità $O(n \log n)$.
11. Consideriamo il problema di calcolare l'espressione $(a^n) \bmod r$ assumendo come input tre interi positivi a, n e r . Progettare un algoritmo che risolva il problema in un tempo $O(\log n)$.