

Laurea in “Informatica”

Corso di “Algoritmi e Strutture Dati”

5 Giugno 2006

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Sono ammessi al più 3 scritti consegnati tra Giugno 2006 e Febbraio 2007.
3. Non è possibile consultare appunti, libri o persone, né uscire dall'aula.
4. Per raggiungere la sufficienza occorrono almeno 3 esercizi risolti senza alcun errore.
5. Le soluzioni degli esercizi devono:
 - a. spiegare a parole l'algoritmo usato (anche con eventuali disegni)
 - b. commentare l'eventuale procedura Pascal (dettagliando il significato delle variabili)
 - c. giustificare la correttezza e tutti i passaggi matematici
 - d. dimostrare la complessità (con equazioni di ricorrenza se necessario)

1. Si scriva la procedura Pascal COMPONENTICONNESSE per determinare le componenti connesse di un grafo non orientato e la si esegua sul grafo $G = (N, A)$, dove $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ e $A = \{[1,4], [1,5], [2,3], [2,6], [3,4], [4,5], [5,6], [7,8], [7,9], [8,9], [10,11]\}$. Si assuma che i vettori di adiacenza siano ordinati in modo crescente e se ne mostri il contenuto.

2. Sia dato un albero binario T in cui ciascun nodo contiene un elemento intero relativo. Dato un nodo v di T , definiamo SOMMAZERO di v la proprietà per cui la somma degli elementi contenuti nel figlio sinistro e nel figlio destro di v , se esistono entrambi, hanno somma uguale a 0 (se esiste un solo figlio di v , allora l'elemento contenuto deve valere 0). Si scriva una funzione Pascal di complessità ottima per determinare se la proprietà SOMMAZERO è verificata per ogni nodo di T , assumendo che T sia realizzato con puntatori.

3. Si definisca, con una descrizione dettagliata, figure, e codice Pascal, una struttura di dati per gestire n chiavi numeriche tali che:

- l'inserimento di una nuova chiave abbia complessità $O(n)$,
- la ricerca del minimo abbia complessità $O(1)$,
- l'estrazione del minimo abbia complessità $O(1)$,
- la cancellazione di un elemento generico abbia complessità $O(n)$.

4. Si scriva la procedura Pascal MERGESORT per ordinare gli n elementi $A[1], \dots, A[n]$ di un vettore. Dimostrarne inoltre la sua complessità per mezzo di equazioni di occorrenza.

5. Dati un array A (NON ordinato) composto da n interi distinti e un intero z , scrivere un algoritmo efficiente (in Pascal) che sia in grado di determinare il numero delle coppie di indici (i, j) con $i < j$ tali che $A[i] + A[j] = z$. Ad esempio, se $A = [3,5,6,1,4,2,8]$ e $z = 9$, il risultato dovrà essere 3 ($1+8, 3+6, 4+5$).

6. Dato un insieme A di n interi distinti e un intero z , si vuole decidere se esiste un sottoinsieme S di A tale che la somma degli elementi in S sia uguale a z . Scrivere (in pseudocodice Pascal) un algoritmo non deterministico che richieda tempo polinomiale.