

## Prova scritta per l'esame di Algoritmi e Strutture Dati con Laboratorio

### Domanda 1 (15 Punti)

Date  $n$  matrici,  $A_1, A_2, \dots, A_n$ , compatibili per il prodotto, ovvero  $(A_1 \times A_2 \times A_3 \times \dots \times A_n)$  è ben definito, si dia un algoritmo di Programmazione Dinamica che calcola il modo ottimo di calcolare tale prodotto. Si dia un'analisi caso pessimo del vostro algoritmo.

### Domanda 2 (15 Punti)

Si definisca un albero 2-3. Si discuta la procedura di ribilanciamento dell'albero nel caso in cui venga violata la condizione 2-3 a causa dell'aggiunta di una foglia. Si valuti il costo dell'intera procedura di ribilanciamento.

### Domanda 3 (20 Punti)

Un'espressione booleana in forma disgiuntiva normale è un OR di clausole, in cui ogni clausola è un AND di letterali. Ad esempio  $F(X_1, X_2, X_3) = (X_1 X_2) + (X_3 X_1)$  rispetta tale definizione.

(A) Si dia un algoritmo polinomiale che, presa in input una formula  $F$  di  $n$  variabili in forma disgiuntiva normale, stabilisca se la formula è soddisfattibile o meno.

(B) Alla luce del fatto che la soddisfattibilità delle espressioni booleane è un problema NP-Completo, si indichi per quale motivo l'algoritmo in (A) non prova che  $P=NP$ .

### Domanda 4 (15 punti)

Si definisca uno Heap, e si applichi la procedura Buildheap al seguente vettore, specificando ogni passo dell'algoritmo

13	21	76	57	54	44	31	6	45	28	10	92	64	30	42
----	----	----	----	----	----	----	---	----	----	----	----	----	----	----

Dove si trova il massimo dell'insieme? Dove si trova il secondo numero più grande?  
Dove si trova il minimo?

### Domanda 5 (15 punti)

Quand'è che una funzione  $f(x)$  è  $O$  (o grande) di una funzione  $g(x)$ ? Cos'è un lower bound?

Si consideri il seguente frammento di algoritmo in pseudo-codice, dove  $A$  è un vettore di taglia  $n$ :

```
For j=1 to n do
    { k=1;
      while k<j do
        A[k]=A[k]+A[k+1];
        k=k+1; }
```

Si calcoli la complessità di tempo dell'algoritmo e si dica che cosa calcola.

**Domanda 6 (20 punti)**

Definire il problema della connettività forte in un grafo orientato e dare un algoritmo per decidere se un grafo è fortemente connesso. Dare brevemente e informalmente una spiegazione della correttezza dell'algoritmo, e la sua complessità.