

008AA – ALGORITMICA E LABORATORIO

Verifica dell'8 aprile 2010

Cognome Nome:

N. Matricola:

Corso: A B

Esercizio 1. (5+4 punti) Data la seguente funzione ricorsiva `foo`, trovare la corrispondente relazione di ricorrenza e risolverla utilizzando il teorema principale.

```
foo( n ){
  if ( n < 5 ) return 1;
  if ( n <= 200 ) {
    temp = 0;
    for ( i = 1; i <= n; i = i + 1 )
      for ( j = 1; j <= n; j = j + 1 )
        for ( k = 1; k <= n; k = k + 1 )
          for ( m = 1; m <= n; m = m + 1 )
            tmp = tmp + i * j * k * m;
    return tmp + foo( n/5 ) + foo( n/2 );
  }
  tmp = 0;
  for ( i = 1; i*i < n; i = i + 1 )
    tmp = tmp + i*i;
  return tmp * foo( n/2 ) * foo( n/2 ) * foo( n/2 ) * foo( n/2 );
}
```

Cognome Nome:

N.Matr:

Esercizio 2. (*6+3 punti*) Sia dato un array A di n interi distinti e positivi. Progettare un algoritmo ricorsivo basato sulla tecnica *divide et impera* che conta il numero di elementi che sono minimi relativi del vettore A (escludendo dal conteggio $A[0]$ e $A[n - 1]$). Si definisce minimo relativo un elemento che è minore dei suoi adiacenti. Per esempio, $A = [3, 4, 2, 8, 7, 10, 12, 15, 14, 20]$ ha tre minimi relativi che sono $\{2, 7, 14\}$.

Si valuti inoltre la complessità in tempo e spazio al caso peggio dell'algoritmo proposto.

Suggerimento: Prestare attenzione agli estremi dei sotto-vettori nelle chiamate ricorsive.

Cognome Nome:

N.Matr:

Esercizio 3. ($4+1$ punti) Una sequenza *bilanciata* di parentesi può essere definita ricorsivamente come segue: la sequenza vuota è bilanciata; se S e T sono sequenze bilanciate, allora $(S)T$ e $S(T)$ sono sequenze bilanciate. Per esempio, $()(())(())$ e $(())(())(())$ sono sequenze bilanciate, mentre $()(())(())()$ e $)()(())$ non lo sono.

Progettare un algoritmo efficiente che, presa in ingresso una sequenza di parentesi, calcola il minimo numero di parentesi da aggiungere per renderla bilanciata. Per esempio, la risposta è 0 per $()(())(())$ e $(())(())(())$ in quanto sono già bilanciate, mentre è 3 per $()(())(())()$ e 2 per $)()(())$. L'algoritmo deve essere spiegato anche a parole.

Valutare inoltre la complessità in tempo al caso pessimo dell'algoritmo proposto.

Cognome Nome:

N.Matr:

Esercizio 4. (*3+4+2 punti*) Scrivere la formula ricorsiva di programmazione dinamica per risolvere il problema della sottosequenza comune più lunga (*LCS*). Applicare tale formula alle due sequenze $X = \text{DAEABCDE}$ e $Y = \text{ZADBCDYE}$, costruendo la relativa tabella di programmazione dinamica. Infine, mostrare sulla tabella come effettuare il percorso a ritroso per trovare una sottosequenza comune più lunga, ed indicarla.