

Corso di Programmazione

II Prova di accertamento del 20 Giugno 2022 / B

cognome e nome

Riporta in modo chiaro negli appositi spazi le soluzioni degli esercizi, oppure precise indicazioni se alcune soluzioni si trovano in un foglio separato. Scrivi inoltre il tuo nome nell'intestazione e su ciascun ulteriore foglio che intendi consegnare.

2. memoization

Una sequenza s di `double` si definisce *smorzantesi* (damping) se ogni suo elemento ha un valore che ricade strettamente all'interno dell'intervallo delimitato dai due elementi precedenti, quando ci sono entrambi. Formalmente:

$$\min(s[i-2], s[i-1]) < s[i] < \max(s[i-2], s[i-1]) \quad \text{per } i \geq 2$$

Data una sequenza s , rappresentata da un array di `double`, il programma ricorsivo riportato nella pagina seguente ne determina la *lunghezza della sottosequenza smorzantesi più lunga* (*llds = length of the longest damping subsequence*).

```

public static int llds( double[] s ) {           // s[i] > 0 per i in [0,n-1], dove n = s.length
    return lldsRec( s, 0, UNKNOWN, UNKNOWN );
}

private static int lldsRec( double[] s, int k, int i, int j ) {

    if ( k == s.length ) {
        return 0;                               // coda di s vuota
    } else if ( ( i == UNKNOWN ) ||
                ((Math.min(s[i],s[j]) < s[k]) && (s[k] < Math.max(s[i],s[j])))) ) {

        return Math.max( 1+lldsRec(s,k+1,j,k),
                        lldsRec(s,k+1,i,j) );    // s[k] può essere scelto o meno
    } else {
        return lldsRec( s, k+1, i, j );        // s[k] non può essere scelto
    }
}

private static final int UNKNOWN = -1;         // indice i indefinito

```

Completa il programma riportato qui sotto, che applica una tecnica *top-down* di *memoization* per rendere più efficiente la computazione ricorsiva avviata da `llds`.

```

public static int llds( double[] s ) {

    int n = s.length;

    ..... mem = ..... ;

    .....

    .....

    .....

    .....

    return lldsRec( s, ..... , mem );
}

private static int lldsRec( double[] s, ..... , ..... mem ) {

    if ( mem ..... == ..... ) {

        .....

        .....

        .....

        .....

        .....

    }

    return ..... ;
}

```