

## Fondamenti di informatica II - esame del 9 luglio 2019

1. Illustrare una riduzione – a vostra scelta - che dimostra che un problema è NP-difficile.
2. Definire le classi P e NP e indicare le relazioni di contenimento fra le classi
3. Dimostrare che se un problema A è riducibile in tempo polinomiale ad un problema B e che B è riducibile in tempo polinomiale a C, allora A è riducibile a C.
4. Nell'analisi di complessità e, in particolare nella notazione  $O()$ , si ignorano le costanti moltiplicative e additive. Discutere questa scelta indicandone i vantaggi e gli svantaggi.
5. Si consideri il seguente programma per il calcolo della più lunga sottosequenza comune a due stringhe X e Y.  
Input: Stringhe X e Y con n e m elementi,  
Output: Per  $i = 0, \dots, n-1$ ,  $j = 0, \dots, m-1$ , la lunghezza  $L[i, j]$  della più lunga stringa che è comune sia alla stringa  $X[0..i] = x_0x_1x_2\dots x_i$  che alla stringa  $Y[0..j] = y_0y_1y_2\dots y_j$   
for  $i = 1$  to  $n-1$  do  
     $L[i, -1] = 0$   
for  $j = 0$  to  $m-1$  do  
     $L[-1, j] = 0$   
for  $i = 0$  to  $n-1$  do  
    for  $j = 0$  to  $m-1$  do  
        if  $x_i = y_j$  then  
             $L[i, j] = L[i-1, j-1] + 1$   
        else  
             $L[i, j] = \max\{L[i-1, j], L[i, j-1]\}$   
return array L  
Eseguire il programma per le stringhe  $X = \text{"GMJTAUZ"}$  e  $Y = \text{"MZJAWGU"}$  fornendo la matrice L calcolata.  
Esprimere il costo del programma in funzione della lunghezza dell'input. Motivare le risposte.
6. Descrivere un analizzatore sintattico di tipo top-down specificando l'input dell'analizzatore, i possibili risultati calcolati e un possibile algoritmo per l'analisi.
7. Descrivere un automa a stati finiti che riconosce il linguaggio formato dalle stringhe binarie di 0 e 1 costituito da tutte e sole le stringhe binarie contenenti un numero pari di 0 e un numero pari di 1 (in qualunque posizione; quindi l'automa accetta le stringhe 0011, 01010101, 00100100 e rifiuta le stringhe 0111 e 00100).
8. Sia data la grammatica  $G = (N, T, P, S)$  con insieme dei simboli non terminali  $N = \{S, A\}$ , insieme simboli terminali  $T = \{a, b\}$ , assioma S e produzioni specificate nel seguito:  
 $S \rightarrow aS$  (1)  
 $S \rightarrow aA$  (2)  
 $A \rightarrow bA$  (3)  
 $A \rightarrow b$  (4)  
Fornire un albero di derivazione della stringa aabbb; specificare il linguaggio generato dalla grammatica e discutere se la grammatica è ambigua o no.
9. Fornire una grammatica non ambigua per generare il linguaggio delle espressioni aritmetiche avente cinque simboli terminali: id (che rappresenta un identificatore), i simboli di operazione + e - e le parentesi tonde ( e ). Ovviamente la grammatica deve generare tutte e sole le stringhe che sono corrette (ad esempio non deve generare la stringa  $id + ((id - id)$  o la stringa  $id ++ id - id$ ).  
Motivare inoltre perché questo linguaggio non può essere generato da una grammatica di tipo 3.