Prof. L. Margara **Scritto del 1/9/2022**

Algoritmi e Strutture Dati (Classe A)

Esercizio 1.[11 punti]
Fornire una funzione in pseudocodice che, preso in input un grafo G orientato (rappresentato con matrice di adiacenza) e una coppia di nodi i e j di G, restituisca 1 se il numero di cicli distinti di lunghezza 8 che passano per il nodo i è maggiore o uguale del numero di cicli distinti di lunghezza 16 che passano dal nodo j, 0 altrimenti. Sia n il numero di nodi di G, la funzione fornita deve avere un costo computazionale $O(n^3)$. Discutere la correttezza dell'algoritmo fornito e determinare il suo costo computazionale. (Suggerimento: utilizzare la moltiplicazione di matrici)

Traccia Soluzione esercizio 1.
Sia M la matrice di adiacenza di G. Prima di tutto calcoliamo M^8 e M^{16}. In queste due matrici ci saranno memorizzati in posizione i, j il numero di cammini distinti dal nodo i al nodo j lunghi 8 e 16, rispettivamente. A questo punto basta confrontare $M^8(i,i)$ e $M^{16}(j,j)$ e il gioco è fatto.

Esercizio 2.[11 punti]
Risolvere in ordine di grandezza la seguente equazione ricorsiva.
$T(n) = \begin{cases} 1 & \text{if } n \leq 1, \\ T(\log(\log(n))) + n & \text{if } n > 1. \end{cases}$

Soluzione esercizio 2.
Svolgendo l'equazione otteniamo:

$$T(n) = n + \log(\log(n)) + \log(\log(\log(\log(n)))) + \cdots$$

Il numero di termini della sommatoria è sicuramente minore di $\log^*(n)$. Quindi

$$T(n) \leq n + \log^*(n)\ \log(\log(n))$$

E quindi $T(n) = O(n)$. Banalmente vale anche $T(n) = \Omega(n)$ e quindi $T(n) = \Theta(n)$

Esercizio 3.[11 punti]
Fornire una funzione ricorsiva in pseudocodice che, preso in input il puntatore ad una lista, elimini tutti gli elementi con key maggiore di 10 e restituisca il puntatore alla lista cosi ottenuta. Ogni nodo deve avere solo due campi: key e next. Discutere la correttezza dell'algoritmo fornito e determinare il suo costo computazionale.

Soluzione esercizio 3.

```
F(p)
if p = nil
    return p
x = F(p.next)
if p.key > 10
    return x
else
    p.next = x
    return p
```