

Esercizio 1 (Distanza massima). Dato un grafo diretto G e un nodo $x \in V(G)$, progettare un algoritmo di complessità $O(n + m)$ che restituisca il numero dei nodi raggiungibili da x che si trovano alla massima distanza.

```
BFS_count_max( $G$ :grafo,  $x$ :nodo) {  
    Dist[ $n$ ] = {-1, -1, ..., -1}  
    Dist[ $x$ ] = 0  
    S:Stack  
    S.push( $x$ )  
    while(S ≠ ∅) {  
         $z$  = S.top()  
        For each  $z \in z.adj$  {  
            if (Dist[ $z$ ] == -1) {  
                Dist[ $z$ ] = Dist[ $z$ ] + 1  
                S.push( $z$ )  
            }  
        }  
        S.pop()  
    }  
     $M$  = max(Dist)  
     $c$  = 0  
    For ( $i$  = 0 ...,  $n-1$ ) {  
        if (Dist[ $i$ ] ==  $M$ ) {  $c$  ++ }  
    }  
    return  $c$   
}
```

Utilizzo un BFS per calcolare le distanze da x agli altri nodi, ciò ha costo $O(n+m)$, per poi contare quanti nodi hanno distanza massima da x , ha costo $O(n)$