

Esercizio 1 (22.2-9, [1]). Fornire un algoritmo in pseudo-codice che, dato un grafo non diretto e connesso $G = (V, E)$, trova una passeggiata in G che attraversa tutti gli archi una e una sola volta in ognuna delle due direzioni in tempo $O(|V| + |E|)$.

Faccio una DFS, ogni volta che considero un nodo già visitato, percorro avanti ed indietro l'arco incidente.

```
DFS(G:grafo, x:nodo, Vis:array, L:lista, prec:nodo) {
    if (x != prec) { L.add( (prec, x)) }
    Vis[x] = 1
    for each y ~ x {
        if (Vis[y] == 0) {
            DFS(G, y, Vis, L, x)
        } else if (y != prec) {
            L.add(x, y)
            L.add(y, x)
        }
    }
    if (x != prec) { L.add( (x, prec)) }
}
```

Esercizio 2 (22.4-2, [1]). Fornire un algoritmo in pseudo-codice che dato un grafo diretto e aciclico $G = (V, E)$ e due vertici s e t , restituisce il numero di tutti i cammini da s a t in G .

Sia $\#P(s, t)$ il numero di cammini da s a t , vale la seguente:

$$\#P(s, t) = \sum_{u \in N} \#P(s, u) \quad \text{dove } N = \{u \mid (u, t) \in E(G)\}$$

```
Cammini(s:nodo, t:nodo, G:grafo) {
    if (t == s) { return 1 }
    K = 0
    for each u | (u, t) ∈ E(G) {
        K += Cammini(s, u)
    }
    return K
}
```

