

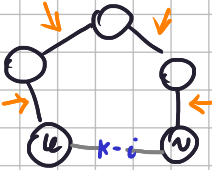
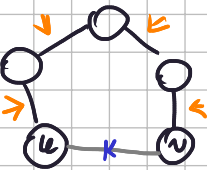
Esercizio 1 (23.1-11, [1]). Sia $G = (V, E)$ un grafo non diretto e connesso con gli archi pesati da una funzione w e T un albero di copertura di G di peso minimo. Supponiamo che il peso di un'arco (u, v) che non appartiene a T diminuisce. Fornire un algoritmo in pseudo-codice che dato G , T e come viene modificato il peso dei un arco (u, v) , trova un albero di copertura di peso minimo nel grafo modificato (senza calcolarlo da capo).

```

Archi_caminno(T: albero, u:nodo, v:nodo) {
    Sol: Stack
    S: Stack
    Vis[u] = {0,0,...,0}
    Vis[u] = 1
    S.push(u)
    while (S != ∅) {
        z = S.top()
        if (z == v) { return Sol }
        if (∃ w ~ z | Vis[w] != 1) {
            Vis[w] = 1
            S.push(w)
            Sol.push((z, w))
        } else {
            S.pop()
            Sol.pop()
        }
    }
}

```

pesano meno di K



possibili
sostituibili
se pesano
piu' di K-i

```

EditEdge ( G: grafo, T: albero, p: Reale) { // p e' il peso di (u,v) modificato
    E = Archi_caminno(T, u, v) // gli archi del cammino (u,v) in T
    me = argmax_{(x,y) ∈ E} (w(x,y))
    if (w(me) ≤ w(u,v)) { return T }
    return {T \ {me}} ∪ {(u,v)}
}

```

Esercizio 2 (23.2-7,[1]). Sia $G = (V, E)$ un grafo non diretto e connesso con gli archi pesati da una funzione w e T un albero di copertura di G di peso minimo. Supponiamo che viene aggiunto un nuovo vertice u e gli archi a se incidenti a G . Fornire un algoritmo in pseudo-codice che dato G , T e la lista di adiacenza di u nei vertici di G , trova un albero di copertura di peso minimo nel grafo $G \cup \{u\}$ (senza calcolarlo da capo).

```

Recalc(G: grafo, T: albero, u: nodo, V: adj di u) {
    me = min arg(w(u,v))
    v ∈ V
    return T ∪ {me} ∪ {u}
}

```