Esercizio 1. (max 6) G un grafo non-diretto e conesso. Dati due vertici v_1 e v_2, la distanza da v_1 a v_2, scritto dist(v_1, v_2), e' la lunghezza minima di un cammino da v_1 a v_2. Invece, dati due sottoinsiemi di vertici X e Y, la distanza da X a Y e

$$\min_{\{x \in X,\, y \in Y\}} \text{dist}(x, y)$$

Si osservi che nel caso in cui $X \cap Y \neq \emptyset$, diciamo che dist(X,Y) = 0.

Dare lo pseudocodice di un algoritmo $O(|V| + |E|)$ per trovare dist(X,Y) dati G, X,Y in input.

```
BFS_set (G:grafo, X:nodi, Y:nodi){
    Q:queue
    Dist[n]={-1,-1...,-1}
    for each x∈X{
        Q.push(x)
        Dist[x]=0
    }
    while(Q≠∅){
        u=Q.pop()
        For each v∈u{
            iF(Dist[v]==-1){
                Dist[v]=Dist[u]+1
            }
        }
    }
    M = ∞
    for each t∈Y{
        M=min(M,Dist[t])
    }
    return M
}
```

Esercizio 2. (max 9) Dare lo pseudocodice per risolve il seguente problema: dati n oggetti $x_1, x_2,\ldots,x_n$ ognuno di un costo c_i e un valore v_i, $1 \le i \le n$, e un vincolo A, trovare il sottoinsieme di $x_1,x_2,\ldots,x_n$ di costo minimo con valore totale al minimo A.

E' un problema di programmazione dinamica, definisco la matrice $T: n \times A$ tale che:

$T[k, \alpha]$ = sottoinsieme di $\{x_1, x_2 \ldots, x_k\}$ di costo minimo e valore $\sum_{i=1}^{k} v_i \ge \alpha$.

Casi banali: $T[0, \alpha]$ = NON DEFINITO $\quad T[k, 0] = \{\}$ // insieme vuoto

$T[1, \alpha] = \{x_1\}$ se $v_1 \ge \alpha$. Ovviamente $0 \le k \le n \ \wedge \ 0 \le \alpha \le A$

```
O = { }
sumV = 0
sumC = 0
For (i = 1, 2 ... n) {
    if (sum < A) {
        O.add(x_i)
        sumV += v_i
        sumC += c_i
    } else {
        toSub = -1
        minC = sumC
        For (j = 1.., i-1) {
            if (sumV - v_j + v_i ≥ A) {
                if (minC - c_j + c_i < minC) {
                    minC = minC - c_j + c_i
                    toSub = j
                }
            }
        }
        if (toSub ≠ -1) {
            SumV += (v_i - v_toSub)
            SumC += (c_i - v_toSub)
            O.remove(x_toSub)
            O.add(x_i)
        }
    }
}
```