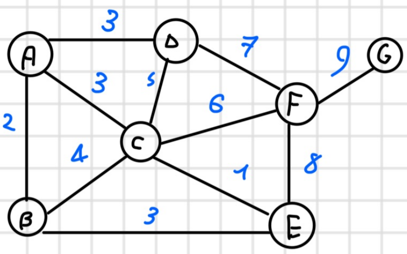
DATO UN GRAFO G CONNESSO, NON ORIENTATO E PESATO, UN MINIMO ALBERO RICOPRENTE DI G E' UN ALBERO RICOPRENTE DI G IN CUI LA SOMMA DEI PESI DEGLI ARCHI E' MINIMA. E' QUINDI UN SOTTOGRAFO DI G TALE CHE:

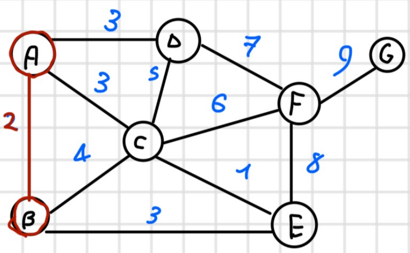
- INCLUDE TUTTI I NODI DI G
- E' UN ALBERO LIBERO, CIOE' CONNESSO E SENZA CICLI.
- LA SOMMA DEI PESI DEGLI ARCHI E' MINIMA.

ESEMPIO DI ESECUZIONE



VISITED = { }

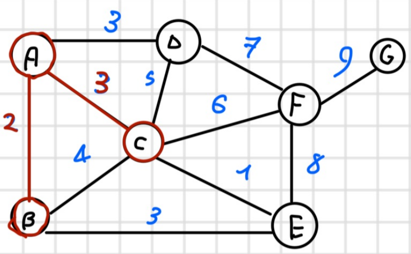
PARTIAMO DA UN NODO QUALSIASI, SCEGLIAMO A. DOBBIAMO TROVARE IL CAMMINO MINIMO DA A A UN ALTRO NODO, IN QUESTO CASO B.



VISITED = {A, B}

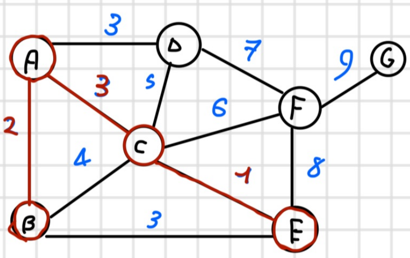
I NODI ROSSI SONO L'INIZIO DEL NOSTRO ALBERO DI COPERTURA MINIMA

ANALIZZIAMO TUTTI I NODI RAGGIUNGIBILI DA A E B, TUTTI I PIU' PICCOLI SONO 3, IN QUESTO CASO BISOGNA SCEGLIERNE UNO A "CASO". SCEGLIAMO C.

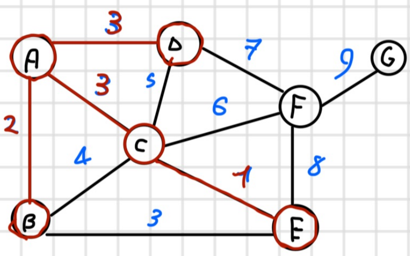


VISITED = { A, B, C }

RIPETIAMO QUESTO PASSAGGIO DI SCEGLIERE IL PIU' PICCOLO ARCO CHE SI COLLEGA A UN NODO NON VISITATO.

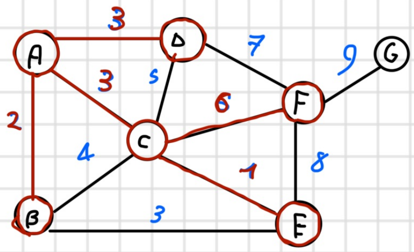


VISITED = {A, B, C, E}

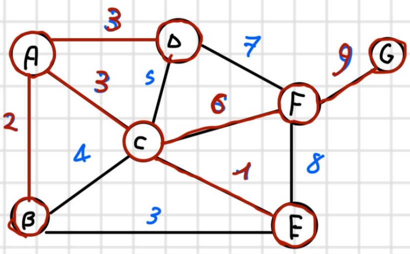


VISITED = {A, B, C, E, D}

A QUESTO PUNTO L'ARCO CON PESO 3 E' IL PIU' PICCOLO MA I VERTICI SONO GIA' NELL'ALBERO (GIA' VISITATI) QUINDI IN QUESTO CASO SCEGLIAMO F:



L'UNICO NODO NON RAGGIUNTO È G. LO AGGIUNGIAMO ALL'ALBERO.



VISITED = {A, B, C, E, D, F, G}

ALGORITMO FINITO

PSEUDO CODICE

```
PRIM (G, S)
    FOR EACH (U NODO IN G) MARCA U COME NON VISITATO   // NECESSARIO
    FOR EACH (U NODO IN G) DIST[U] = ∞
    PARENT[S] = NULL ;  DIST[S] = 0
    Q = HEAP VUOTO
    FOR EACH (U NODO IN G)  Q.ADD (U, DIST[U])
    WHILE (Q NON VUOTA)
        U = Q.GETMIN()
        MARCA U COME VISITATO (NERO)
        FOR EACH ((U,V) ARCO IN G)
                IF (V NON VISITATO && C_{U,V} < DIST[V])
                        PARENT[V] = U ;  DIST[V] = C_{U,V}
                        Q.CHANGE PRIORITY (V, DIST[V])
```

COMPLESSITÀ

LA STESSA DI DIJKSTRA QUINDI $O((m+n) \log n)$