

Minimale spenntreer algoritmer

ALGORITHM: PRIM'S ALGORITME FOR MINIMALE SPENNTRÆR

Input: En sammenhengende, vektet, urettet graf $G = (V, E)$ med vektfunksjon w

Output: Et minimalt spenntre for G

Procedure Prim(G)

```
queue ← empty priority queue
parents ← empty map
Insert(queue, (null, s)) with priority 0, for some arbitrary  $s \in V$ 
while queue is not empty do
    ( $p, u$ ) ← RemoveMin(queue)
    if  $u \notin$  parents then
        parents[ $u$ ] ←  $p$ 
        for ( $u, v$ )  $\in E$  do
            Insert(queue, ( $u, v$ )) with priority  $w(u, v)$ 
return parents
```

- Kjøretidskompleksiteten er den samme som Dijkstra: $\mathcal{O}(|E| \cdot \log(|V|))$

Spenntreer algoritmer

Prims algoritme

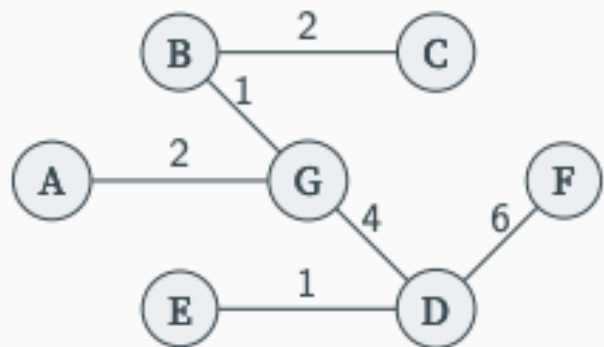
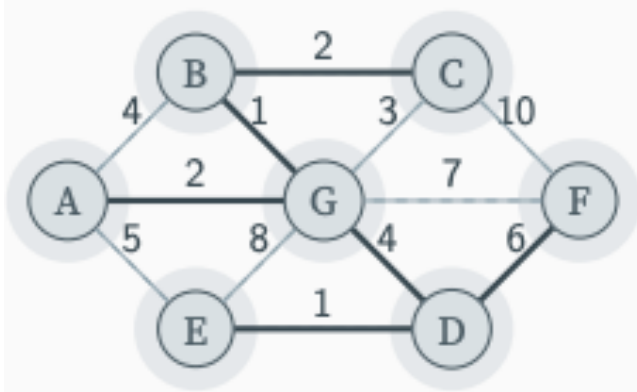
Lage et spenntre med å starte på en vilkårlig node og koble sammen nodene med "billigste" kant som er tilgjengelig/kobla til spenntreet, uten at det blir en sykel

Kruskals algoritme

Lage et spenntre med å se på de billigste kantene og koble sammen nodene som er på hver side av den kanten, helt til du har et spenntre med alle nodene

Boruvkas algoritme

Tar utgangspunkt i komponenter, der komponentene er noder. Ser på komponentene fra A til x og kobler til den "billigste" kanten til komponenten man ser på.



Boruvkas (eksempel)

