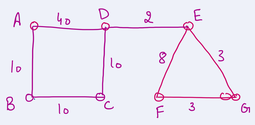
Lecture30

Saturday, 6 June 2020 6:48 PM




bft → Covering all the vertices. → bft.

↳ A B D C E F G ⇒ DE is present.

A B D C E F G ⇒ DE is not present.

dft → A B C D E F G ⇒ B path.

A D E G F C B ⇒ D path ×

A D C B E F G ⇒ DE is not present.

```
public void bft(){
    HashMap<Vertex, Boolean> visited = new HashMap<>();
    Queue<Vertex> q = new LinkedList<>();
    Collection<Vertex> vtces = this.vtces.values();
    for (Vertex vtx : vtces) {
        if (!visited.containsKey()) {
            q.add(vtx);
        }
    }
    while (!q.isEmpty()) {
        Vertex rv = q.remove();
        if (!visited.containsKey(rv)) {
            visited.put(rv, true);
            Syso(rv.name);
            Set<nbrs> {
        }
    }
}
```

⇒ A, B, C, E

| Vertex | Boolean |
|---|---|
| A | true. |
| B | true. |
| D | true. |
| C | true. |
| E | true. |
| F | true |
| G | true |

⇒ Com

⇒ map

vtces.size() == visited.size()

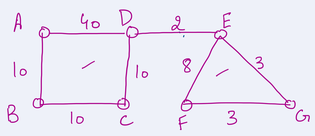
q ⇒ ~~A~~ → ~~B~~ → ~~D~~ → ~~C~~ → ~~E~~ → ~~F~~ → ~~G~~

A B D C E F G

q ⇒ E

DE is not present.

dft → Stack.




isConnected ?

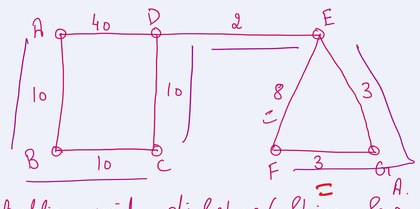
DE → present → true.

DE → not present → false.

Connected Components

DE → [ [A, B, C, D, E, F, G] ]

DE → × [ [A B C D], [E F G] ]




Djikstra's Algo

(A) ↳ Cost is Minimum.

```
public void djikstras(String Source){
    int[] dis = new int[this.vtces.size()];
    for (int i=0; i< dis.length; i++){
        dis[i] = Integer.Max_Value;
    }
    Vertex v = this.vtces.get(Source); // A.
    dis[Source.charAt(0)-65] = 0;
    Queue<String> q = new LL<>();
    q.add(Source);
    while (!q.isEmpty()){
        String str = q.remove();
        Vertex vtx = this.vtces.get(str);
        Set<Vertex> nbrs = vtx.nbrs.keyset();
        for (Vertex nbr : nbrs){
            int oldCost = dis[nbr.name.charAt(0)-65];
            int newCost = dis[vtx.name.charAt(0)-65] + vtx.nbrs.get(nbr);
            if (newCost < oldCost){
                dis[nbr.name.charAt(0)-65] = newCost;
                q.add(nbr.name);
            }
        }
    }
}
```

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 10 | 20 | 30 | 32 | 38 | 35 |

⇐ min Cost

q ⇒ "A" → "B" → "D" → "C" → "E" → "D" → "F" → "G" → "E" → "F" → "F" → "G" → "F"

∞

str ⇒ E, F, G

nbrs ⇒ E, E ⇒ E & G

oldCost ⇒ 40

newCost ⇒ 35 ~~32~~ + 3 = 38

BFt & DFt

1 lec → monday → Questions