

Dijkstra

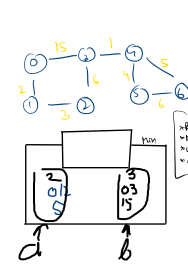


0. $\{ [2, 2] [3, 15] \}$
1. $\{ [0, 2], [2, 3] \}$ → Pair class
2. $\{ [1, 3], [3, 6] \}$ → L node
3. $\{ [0, 15], [2, 6], [4, 5] \}$ → L node
4. $\{ [0, 15], [2, 6], [4, 5] \}$ → L node
5. $\{ [0, 15], [2, 6], [4, 5] \}$ → L node
6. $\{ [0, 15], [2, 6], [4, 5] \}$ → L node
7. $\{ [0, 15], [2, 6], [4, 5] \}$ → L node

*

2 3

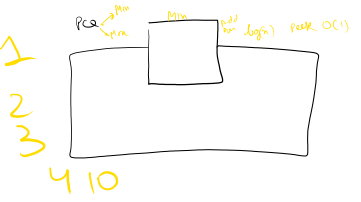
Dijkstra BFS → PA
single source shortest path
to every node in terms
of weights



$E \log(V)$

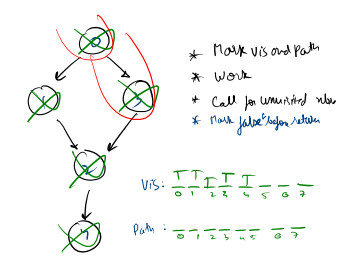
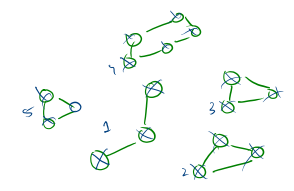
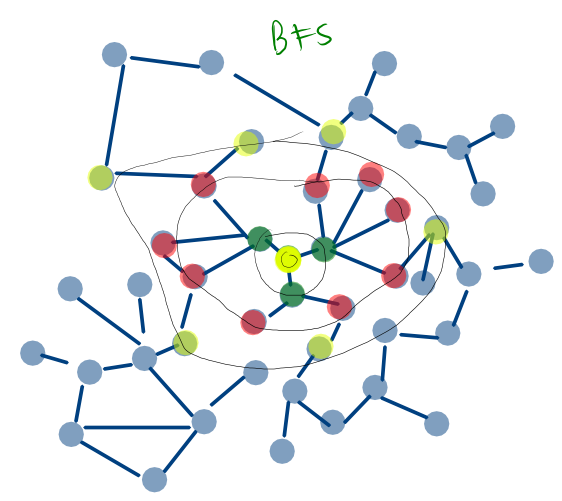
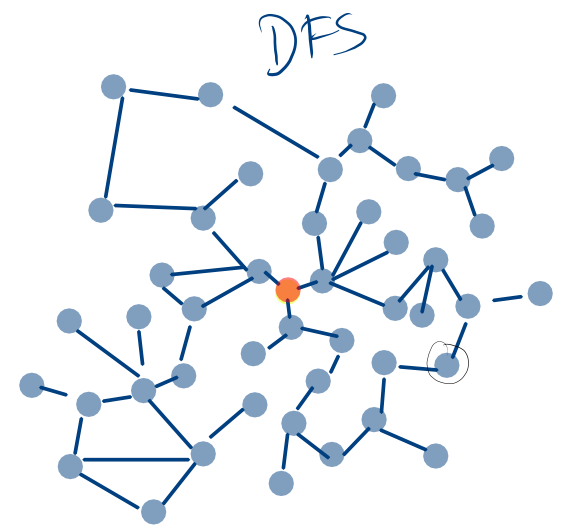
0-0 0 0
1-0 1 2
2-0 1 5
3-0 1 3 11
4-0 1 3 12
5-0 1 3 16
6-0 1 3 17

0, 2, 5, 11, 12, 16, 17



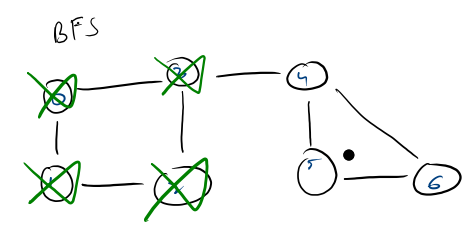
for (ArrayList<Integer> edge: graph.get(curr_node)) {
 ...
}

1: $\{ [4, 5], [5, 6], [6, 7] \}$



Cycle Detection in Undirected Graph

remove
mark
work
Add unvisited
nbrs



vis: $\frac{1}{0} \frac{1}{1} \frac{1}{2} \frac{1}{3} \frac{1}{4} \frac{1}{5} \frac{1}{6}$

2

4

