

In BFS:

How do we extract shortest
Path from parent[]?

First define

Predecessor subgraph: $T = (V_p, E_p)$

$$V_p = \{ x \in V(G) \mid P[x] \neq \text{nil} \} \cup \{s\}$$

$$E_p = \{ (P[x], x) \mid P[x] \neq \text{nil} \}$$

T_p is a shortest Path Tree.

see $\text{PrintPath}(G, s, x)$

recursive algorithm

Depth First Search (DFS)

vertex attributes: $x \in V(G)$

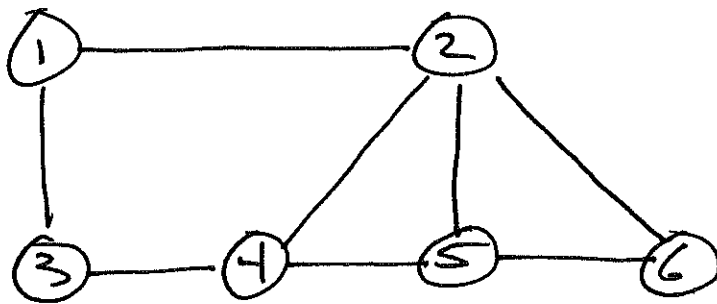
- $\text{color}[x]$: w, g, b
- $P[x]$: Parent or Predecessor
- $d[x]$: discover time
- $f[x]$: finish time

• helper function: $\text{Visit}(x)$
(recursive)

• time variable: $\text{time} \in [0, 2n]$
where $n = |V(G)|$.

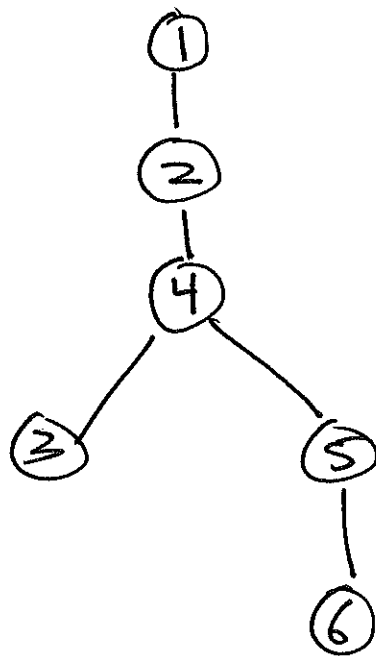
see: $\text{DFS}(G)$
 $\text{Visit}(x)$

Ex.



	adj	color	d	f	p
✓ 1	<u>2</u> <u>3</u>	w g b	1	12	n
✓ 2	<u>1</u> <u>4</u> <u>5</u> <u>6</u>	w g b	2	11	x 1
✓ 3	<u>1</u> <u>4</u>	w g b	4	5	x 4
✓ 4	<u>2</u> <u>3</u> <u>5</u>	w b	3	10	x 2
✓ 5	<u>2</u> <u>4</u> <u>6</u>	w g b	6	9	x 4
✓ 6	<u>2</u> <u>5</u>	w g b	7	8	x 5

DFS Forest:

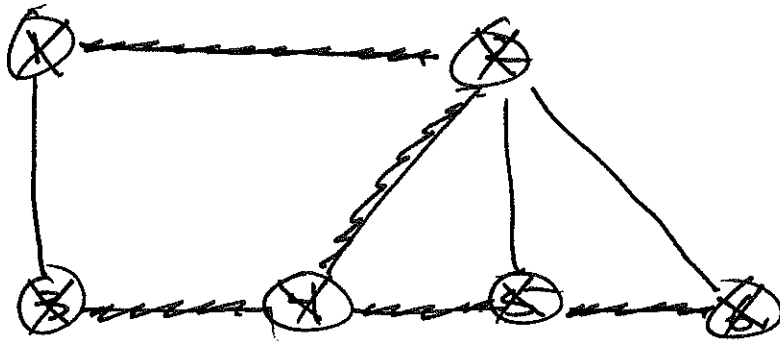


time

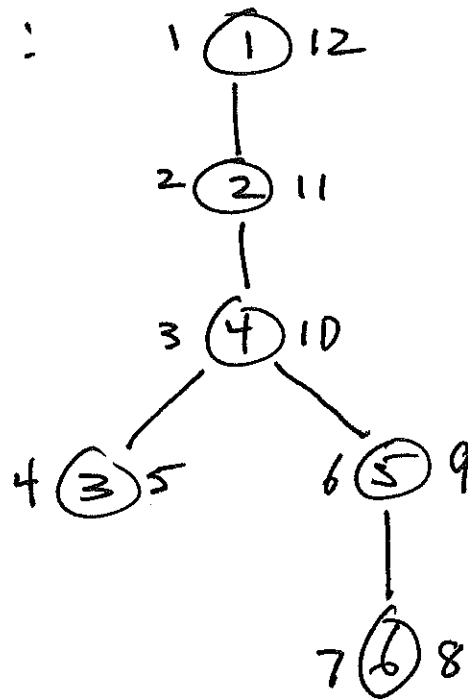
0
 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12

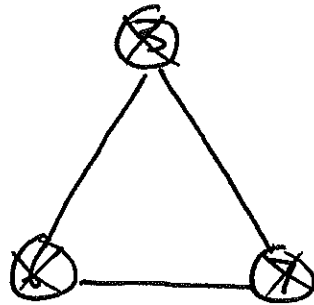
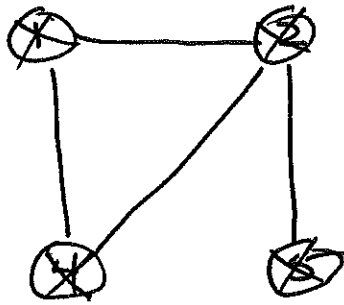
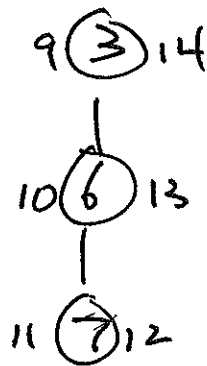
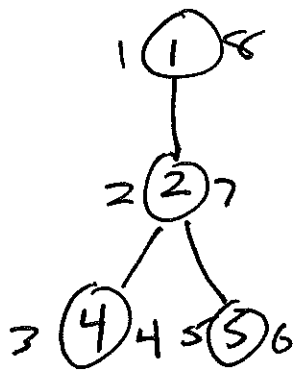
Run DFS "visually"

Ex



DFS forest :

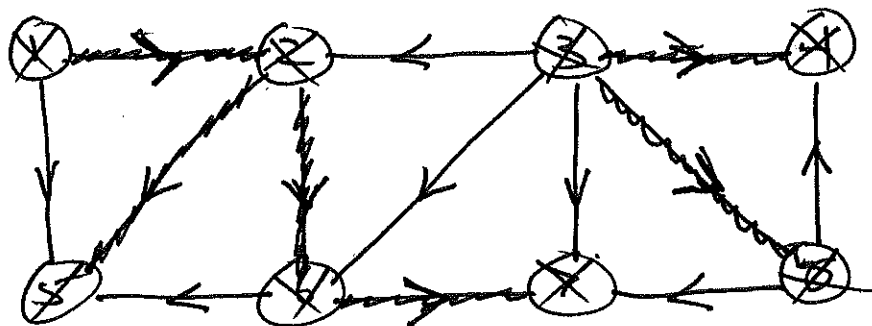


Ex.DFS forest:

	adj	color	d	f	p
1	2 4	b	1	8	n
2	1 4 5	b	2	7	1
3	6 7	b	9	14	n
4	1 2	b	3	4	2
5	2	b	5	6	2
6	3 7	b	10	13	3
7	3 6	b	11	12	6

Ex.

7



Dfs forest:

