Julia Nelson

Homework 3: Graph Algorithms

"I pledge my honor that I have abided by the Stevens Honor System" Julia Nelson

①

Indexes	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	1	0
6	0	0	0	0	0	1	0	1	0	0
7	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0	1
10	0	0	1	0	1	0	0	0	0	0

②

1 → 2 → 4

2 → 5

3 → 5

4 → 2

5 → 4 → 9

6 → 6 → 8

7 → 5

8 →

9 → 7 → 10

10 → 3 → 5

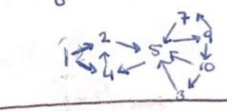
③ Breadth-First Vertices:

1
2
4
5
9
7
10
3

6
8

$6 \to \overset{6}{8}$

8 ← 6 2

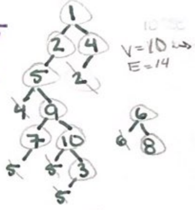


$1 \to \overset{2}{4} \to 5$

$5 \to 9 \to \overset{7}{10}$

10 → 3

$6 \to \overset{6}{8}$

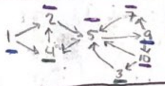


V = 10 → 16 visited
E = 14

E = 7

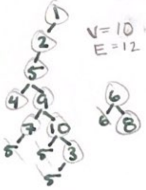
$$\boxed{\begin{matrix}1,2,4,5,9,7,10,3\\ 6,8\end{matrix}}$$

④ Depth-First Vertices:



1 2 5 4 9 7 10 3

8 ← 6 6 8



V = 10 → 14 visits
E = 12

E = 8

$$\boxed{\begin{matrix}1,2,5,4,9,7,10,3\\ 6,8\end{matrix}}$$

⑤ a) RunTime BFS:

Adjacency Matrix:

~~N~~ # vertices

Checks $(V_i, V_j) \in$ in edges

is a V^V memory
check edge O(1)
V^2 space

$\boxed{= O(V^2)}$

b) Adjacency List:

given neighbor vertices
hence # edges → |E|
iterates through vertice, and then # edges an continues to next vertice

$\boxed{= O(|V|+|E|)}$

⑥ RunTime DFS:

a) Adjacency Matrix:

Follows same concept

$\boxed{= O(V^2)}$

b) Adjacency List:

Follows same as BFS

$\boxed{= O(|V|+|E|)}$

DFS tends to ~~&~~ perform better because differing V and E values
#V higher BFS

⑦ Adjacency List is more efficienct because the vertex is listed with its adjacent vestex. This allows it to directly check the vertex/edges. Meanwhile, the Adjacensy Matrix must read through all V×V possible edges to check if it exists taking much more time. (while E<V)

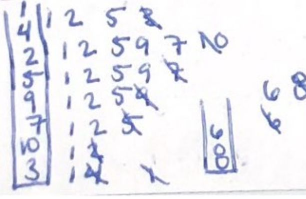
⑧ While using a BFS on an undirected graph to detect if it contains a cycle, you run through a Breadth-First traversal on the graph. While going through visited vertexes if there is an adjacent vertex to the one in question that has already been visited, and this adjacent is not the parent of the one in question, there exists a cycle.

⑨ The DFS is always faster at finding a cycle in an Undirected graph. It is because it avoids processing a vertix more than once through marking it visited or not in an array (And allowing back tracking). This allows for a faster running time than BFS that can take more memory by its booleen array tracking if we visited each + every vertex.

⑩ Topological sort would not work because of the cycles. In the method of topilogical sort, a person looks for the vertex with no "In" edges to start. During this graphs search, it fails in multiple parts. After sorting only 1, the following vestices remaining there are no vestices without any "In" edges. Also, in the ⑧←⑥ part of the graph, there is no start vestex without an "in" edge. Topological sorts work for linear graphs with no cycles where every vestex v comes befor its adjacent vertex u.

⑪

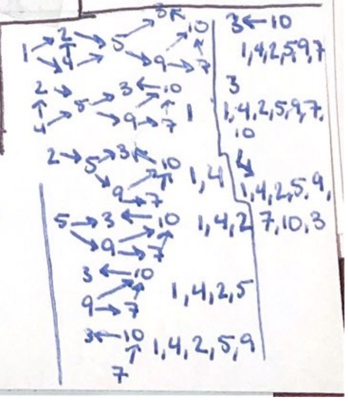
1,4,2,5,9,7,10,3
6,8



⇨ 1,4,2,5,9,7,10,3
6,8

8←6

6,
8
6,8


1
2
4
5
3
10
9
7
3←10
1,4,2,5,9,7
1
3
1,4,2,5,9,7,10
1,4
1,4,2,5,9,
7,10,3
1,4,2
1,4,2,5
1,4,2,5,9