COP 3503 10/13/26

Expected

Lowest Avg on Test: #2, #4, #7

$2^k > n! \rightarrow n = 6$

$2^k > 720$

$\boxed{k = 10}$

testing
Subtraction of eqn
geo series sum

~73 A line

~50 B line

~35 C line

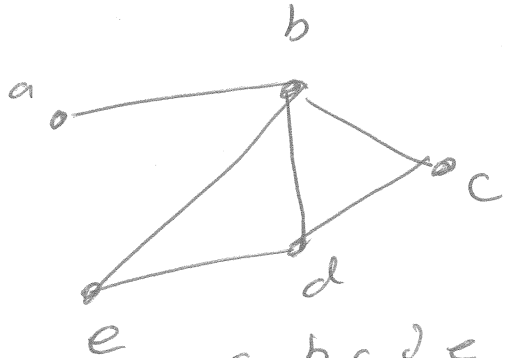
Graphs

1) Definition, 1.5) How to store a graph
2) Depth First Search
3) Breadth First Search → implicit graph
4) Topological Sort

$G = (V, E)$

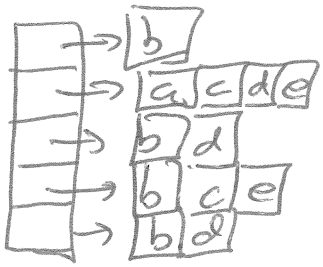
vertices = dots

edges = connection btw pairs of vertices



UNDIRECTED, UNWEIGHTED

	a	b	c	d	e
a	0	1	0	0	0
b	1	0	1	1	1
c	0	1	0	1	0
d	0	1	1	0	1
e	0	1	0	1	0

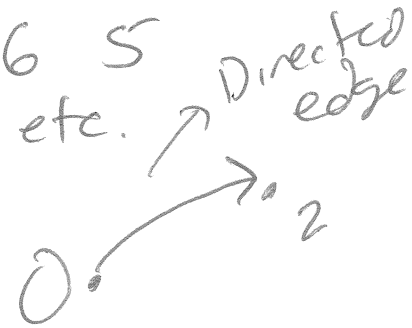


n e

100 1000

```
ArrayList<Integer>[] graph =
        new ArrayList[n];
for (int i=0; i<n; i++)
    graph[i] = new ArrayList<Integer>();
for (int i=0; i<e; i++) {
    int u = stdin.nextInt();
    int v = stdin.nextInt();
    graph[u].add(v);
    graph[v].add(u);
}
```

0 2
1 3
6 5
etc.

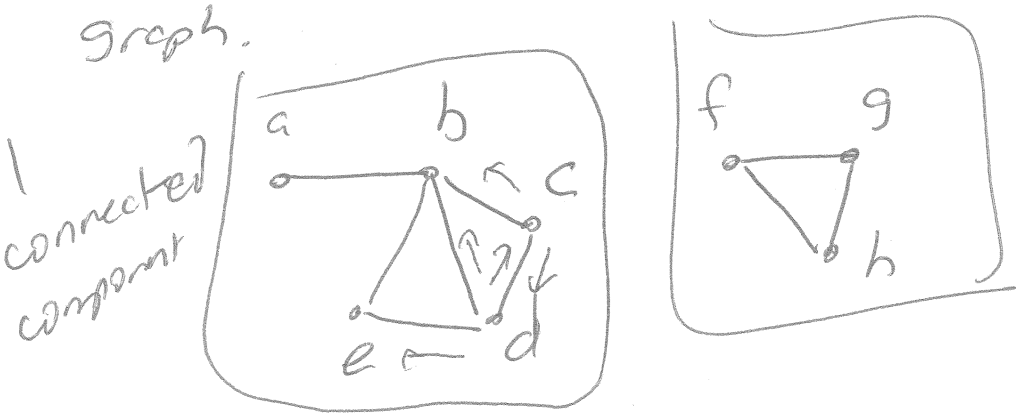


Path is any sequence of vertices connected by edges: a→b→d→c→b→e→d

Simple path: No repeated vertices.

DFS (Depth First Search)

Algorithm to visit all the nodes in a connected portion of an undirected graph.

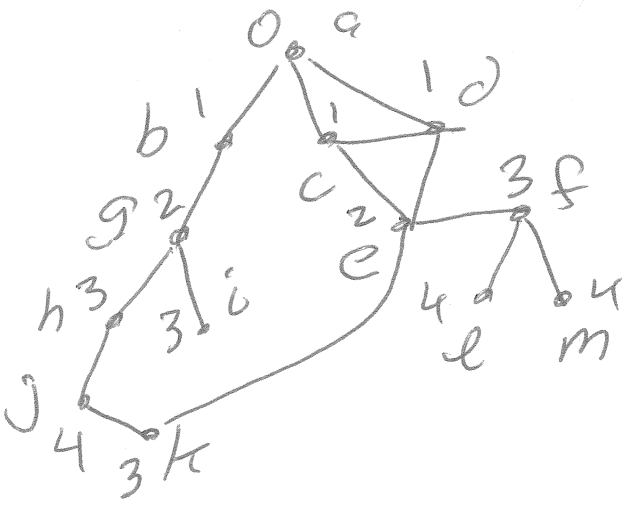


```
dfs(g, v, used):
    used[v] = true
    for (int nextV: g[v])
        if (!used[nextV])
            dfs(g, nextV, used)
```

Run-time $O(V+E)$

DFS(a) → DFS(b) → DFS(c) → DFS(d) → DFS(e)

Breadth First Search




```
BFS(a)
queue: add source
       a
while (q.size() > 0){
    int v = q.poll();
    for (int nextV: g[v]){
        if (dist[nextV] != -1)
            continue;
```

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	1	1	2	3	2	3	3	4	3	4	4

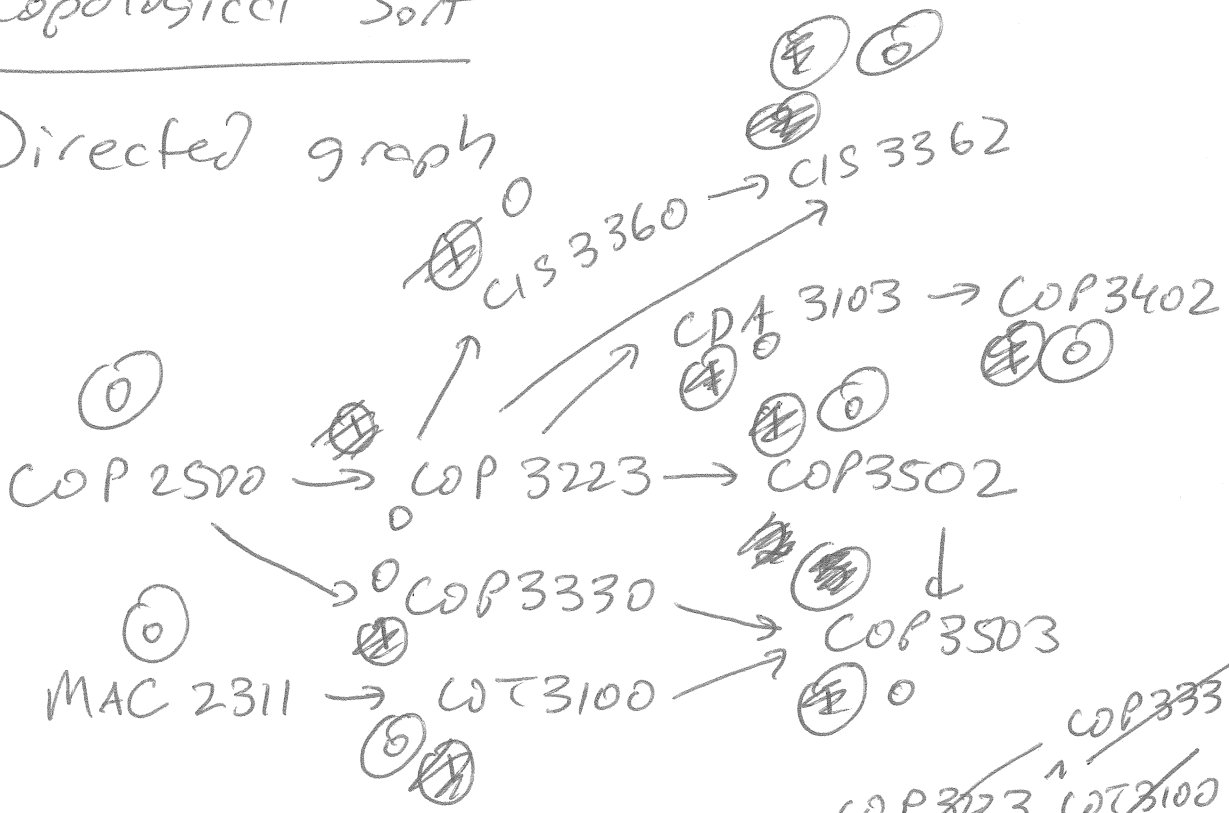
dist ~~= 0~~ 1 2 3 4

queue: ~~a~~ ~~b~~ ~~c~~ ~~d~~ ~~g~~ ~~e~~ ~~h~~ ~~i~~ ~~f~~ ~~k~~ ~~j~~ ~~l~~ ~~m~~

```
        dist[nextV] =
            dist[v] + 1;
    }
}
```

Topological Sort

Directed graph



Queue: COP2500, MAC2311, COP3223, COT3100, CIS3360, CDA3103, COP3502, COP3330, CIS3362, COP3402, COP3503

(add everything in degree 0)

```
while (queue.size() > 0) {
    int cur = queue.poll();
    schedule.add(cur);
    for (int next : g[cur]) {
        indegree[next]--;
        if (indegree[next] == 0)
            queue.offer(next);
    }
}
```

Schedule

2500
2311
~~3223~~
3330
3100
3360
3103
~~3362~~
3502
3362
3402
3503