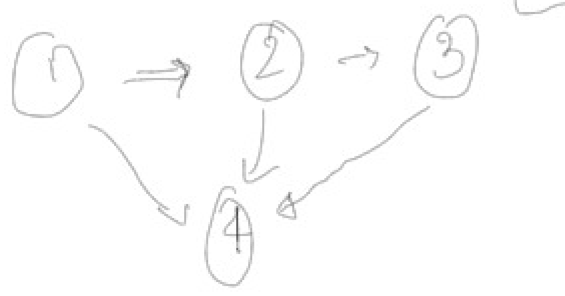
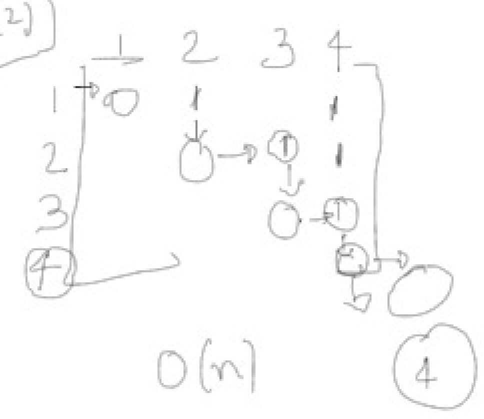
Week 17

Universal Sink

→ OE = 0 , IE = V - 1

→ Adj Matrix → O(n²)





O(n)

sink (4) → O(n)

```
elimination ( )
{   i = 0, j = 0
    while (i < vertices && j < vertices)
    {
        if (a[i][j] == 1)
            i++;
        else j++;
    }
    if (i > vertices)
        return false
    else  check - sink(i)
}
```

O(n)

```
check - sink (int i)
{
  for (int j = 0 ; j < vertices ; j++)
  {
    if (a[i][j] == 1)
        return false
    if (a[j][i] == 0 && i != j)
  }     return false
  return true;
}
```

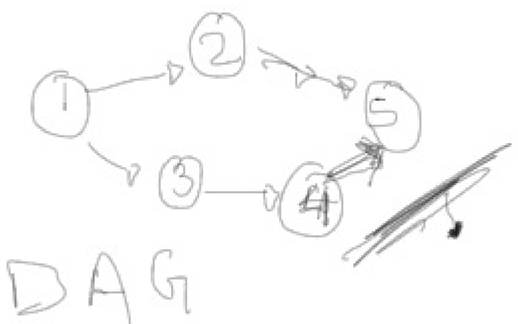
O(n)

O(2n) → O(n)

Topological Sorting

→ Linear ordering for every edge (u, v) u will come before v in the order.

u → v

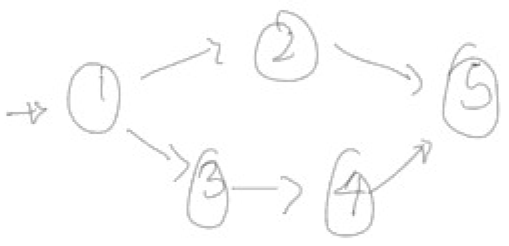


1 2 3 4 5 ✓

1 3 2 4 5 ✓

→ DAG
Directed
Acylic

1 . 2 3 5 → 4 ×



visited → O(V + E)

→ O(V)

| 1 | 3 | 4 | 2 | 5 |

1
3
4
2
5

```
topological sort ( )
{ for(i=0;  i<vertices;  i++)
    if (visited [i] == 0)
        dfs(i, graph, stack)
  }
  while (!stack.empty ( ))
  { cout<< s.top();
      s.pop(); }
```

```
dfs (i, graph, stack)
{       visited [i] = true;
   for(j=0; j< graph[i].size(); j++)
        if ( visited[graph[i][j]]==0)
              dfs(j, graph, stack)
   }
   stack. push(i);
}
```

Alien Dictionary

a → b → c → d → e - - - -

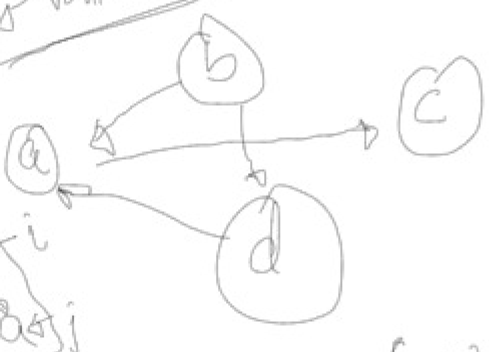
n=4

b a a
a b c d
c a b
c a d

K = 4 → vertices

b a a
a b c d → i
a b c a → j
c a b
c a d

graph[w1[i]].push(w2[j]);



b d a c

b
d
a
c