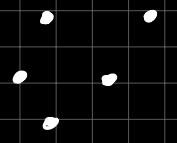
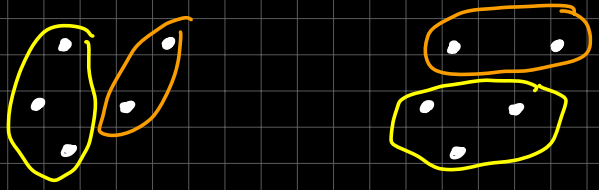
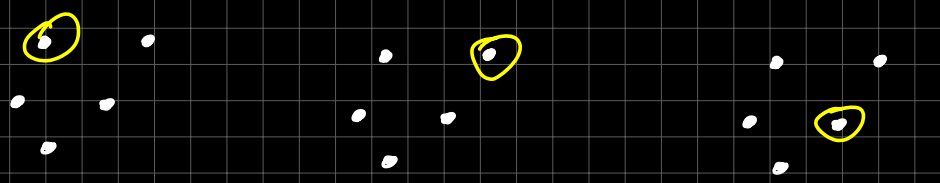
1)

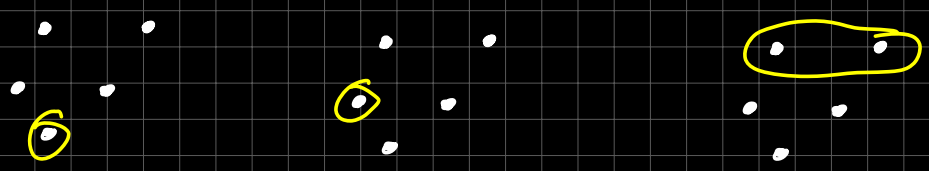


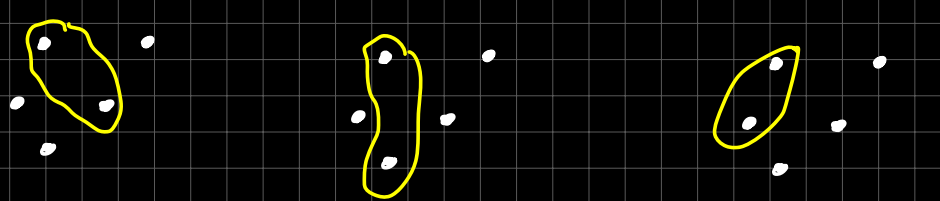
$\overline{K_5}$

- How does the empty set of $K_5$ only have 2 possible bipartite groupings









$$\binom{5}{1} + \binom{5}{2}$$

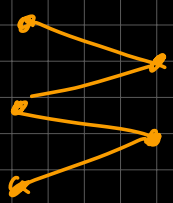
15

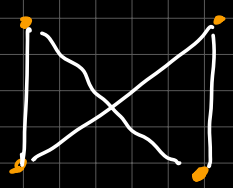
```
int total = 0;
int x = 1;
if (n % 2 == 0)
    while (n/x != x)
        total += n choose x;
```

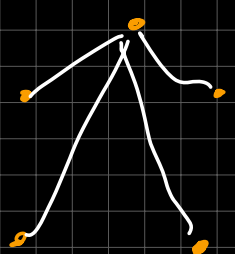
```
else
    while (n/x + 1 != x)
        total += n choose x;

return total;
```

$$\frac{5}{1} + \frac{4}{2} + \frac{3}{3}$$

$$5 + 2 + 1 = 8$$



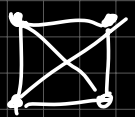




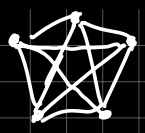
the complement of a disconnected graph is a connected graph

$K_n$ $|E|$ (edge set cardinality)

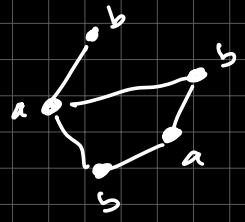
$\Delta = 3 = n$

 $= 6$ $n = 4$

$$\frac{n(n-1)}{2} = \binom{n}{2}$$

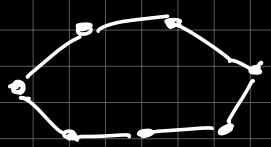


$= 10 \quad n = 5$



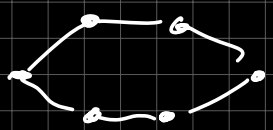

= bipartite

$|E|$ for a cycle



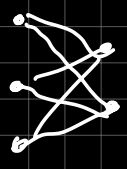
$n = 7$

$C_7$



$|E|$ for $C_n = n$

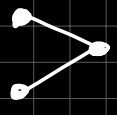
$K_{3,2}$



$|E| = 6$

$|V| = 5$

$K_{2,1}$

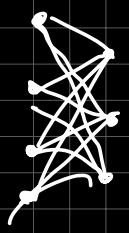


$|E| = 2$

$|V| = 3$

~~$|E|$ of $K_{m,n} = m+n+1$~~

$|V|$ of $K_{m,n} = m+n$



$n \cdot m = |E|$

$|V|$ is always true

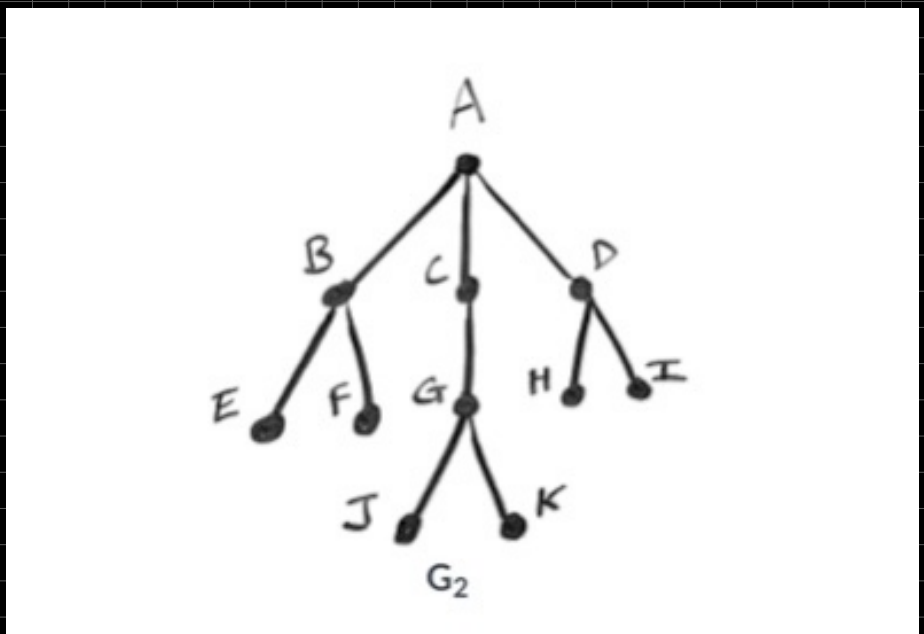
$$\binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4}$$

$$5 + 10 + 10 + 5$$

~~30~~

25

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 | 0 |



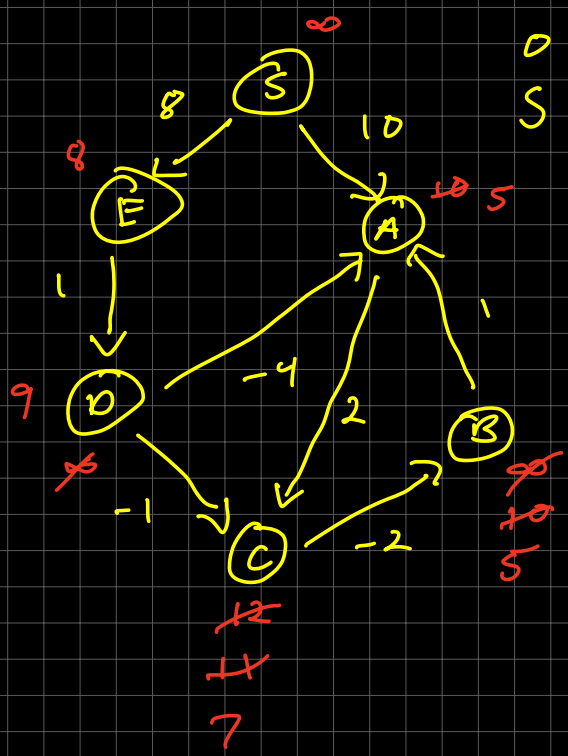
A B E F C G J K D H I

A → B → C → D

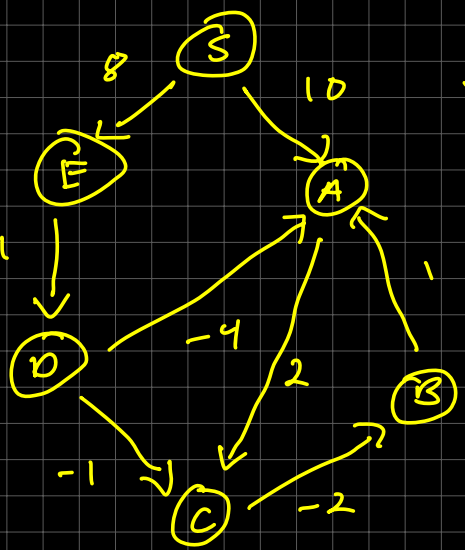
B → E → F

C → G → J → K

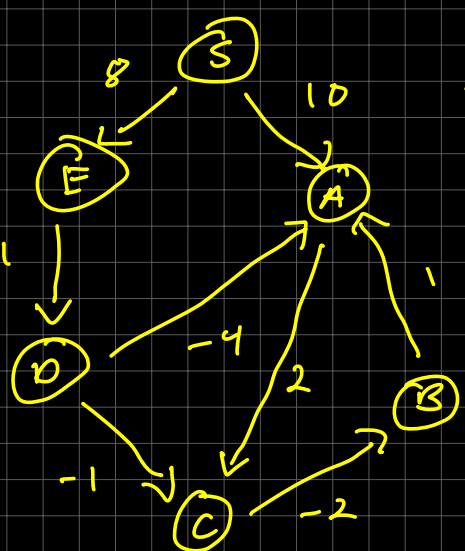
D → H → I



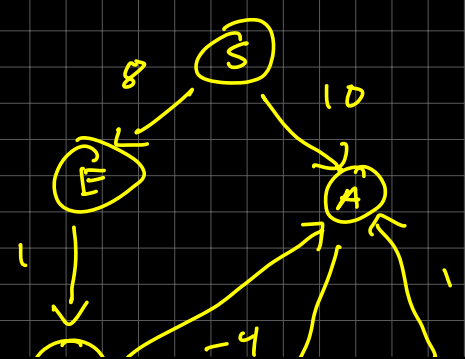

| S | A | B | C | D | E |
|---|---|---|---|---|---|
| 0 | ∞ | ∞ | ∞ | ∞ | ∞ |



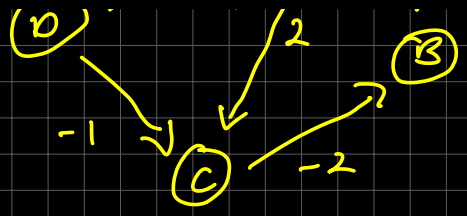

| 0 | 10 | ∞ | 12 | ∞ | 8 |
|---|---|---|---|---|---|
| S | A | B | C | D | E |

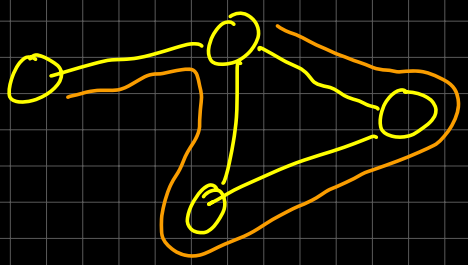



| 0 | 10 | 10 | 12 | 9 | 8 |
|---|---|---|---|---|---|
| S | A | B | C | D | E |




| 0 | 5 | 10 | 7 | 9 | 8 |
|---|---|---|---|---|---|
| S | A | B | C | D | E |

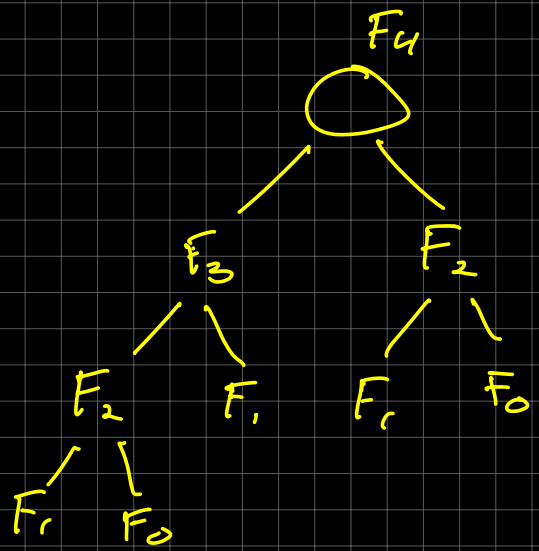


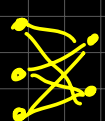



5

15

$2^3 - 1 = 7$



6 edges

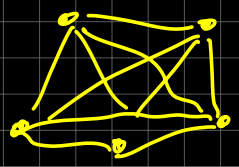
$3 \times 2$

$m \times n$

$$\sum_{i=0}^{abs(n-m)} i$$

$$\sum_{i=0}^{m} i + \sum_{i=0}^{n} i = m^2$$

$$\sum_{i=0}^{m} n \quad \text{or} \quad \sum_{i=0}^{n} m$$



$\frac{5^2-5}{2} = \frac{25-5}{2} = 10$

$|E|(K_5) = \frac{5^2-5}{2} = \frac{25-5}{2} = 10$

$\frac{n^2-n}{2}$ basecase $n=2$

$\frac{4-2}{2} = 1$

Prove that $O(|E| + |V|\log|V|) \leq O(|V|^2)$

3 cases ① ② ③

① $|E| < |V|\log|V|$ ② $|E| = |V|\log|V|$ ③ $|E| > |V|\log|V|$

① + ② : $O(|V|\log|V|)$ which is $\leq O(|V|^2)$

therefore $O(|V|^2)$ is a valid upper bound for all $|V| \geq 2$

③ maximally connected graph with $V$ vertices : $K_n$ $n = |V|$

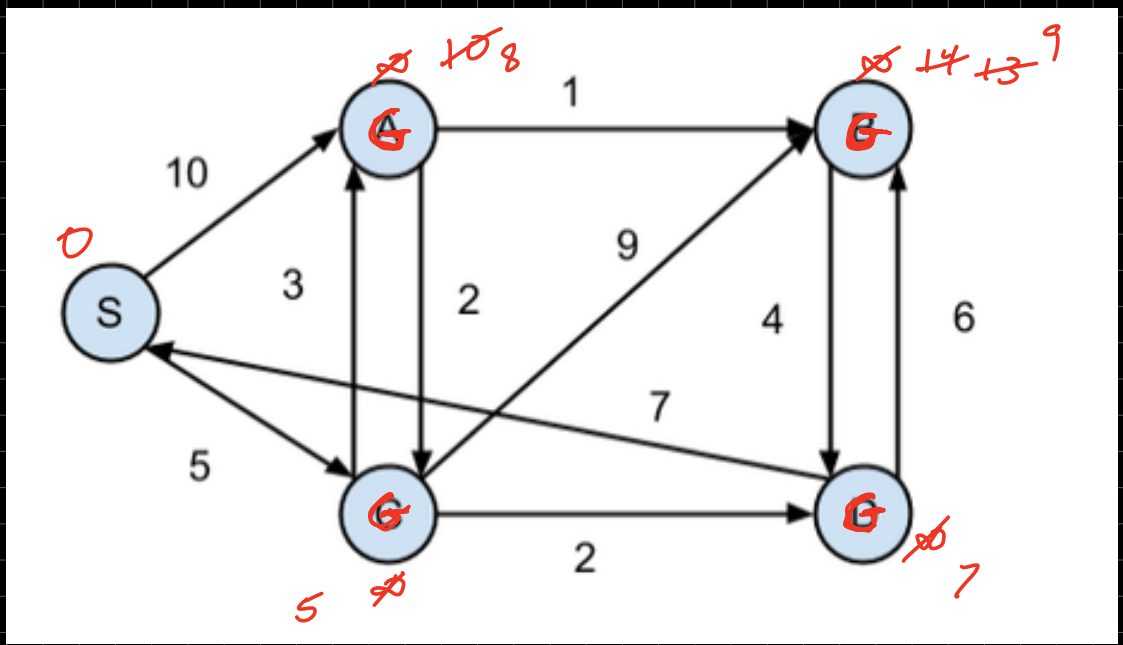
results in largest $|E|$ being $\frac{|V|^2-|V|}{2}$

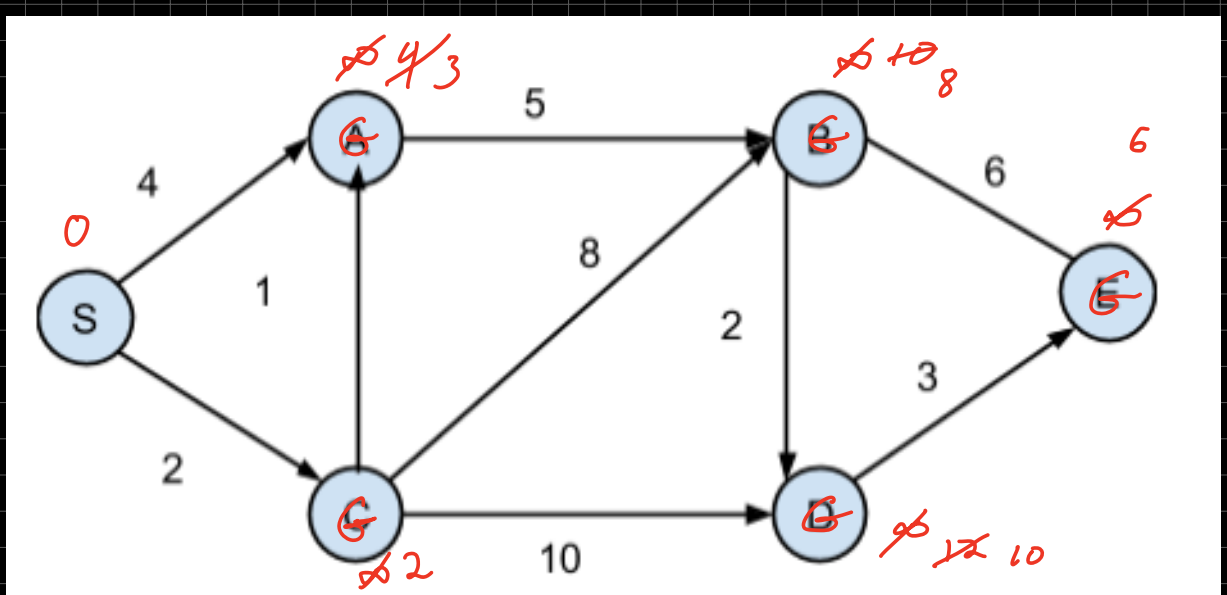
for all $|V| \geq 2$ : $\frac{|V|^2-|V|}{2} \leq |V|^2$

Thus $O(|E|) \leq O(|V|^2)$ for all $|V| \geq 2$

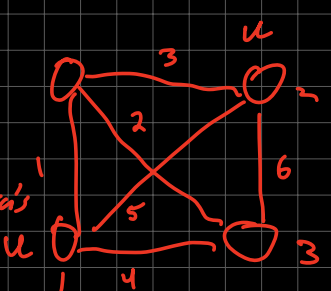
& $O(|V|^2)$ is a valid upper bound

# Dijkstra's Practice




Trying to figure out space complexity of Queue in Dijkstra's

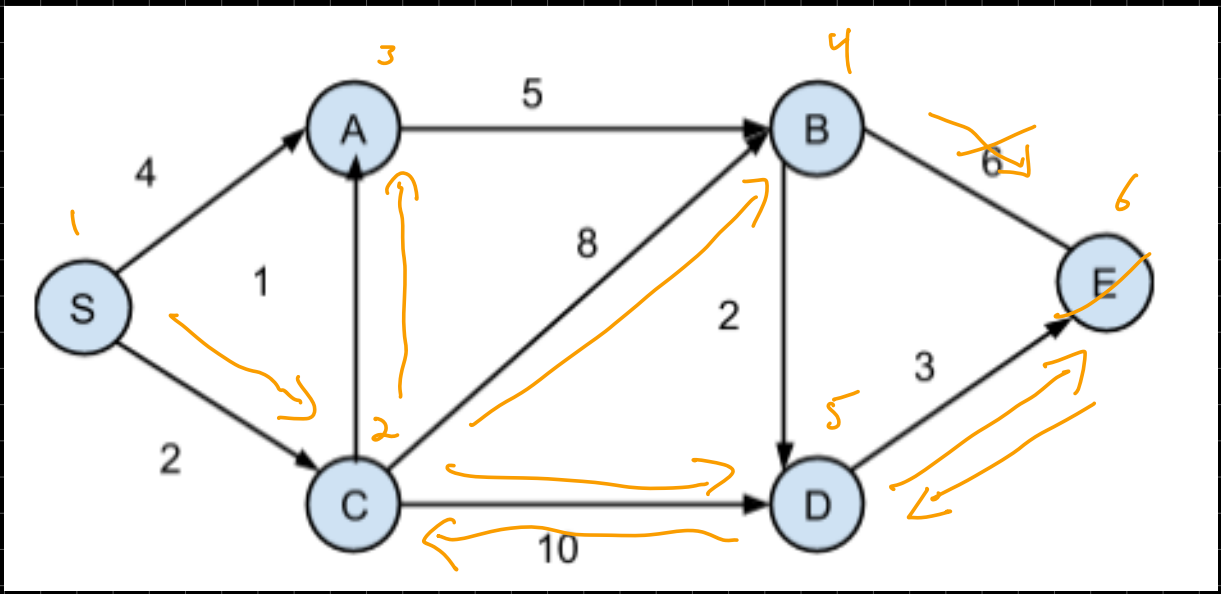



Q: 1, 3, 2

Q: 3, 2, 3, 2

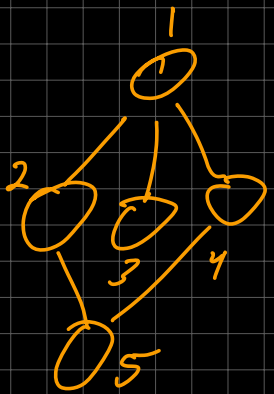
Q: 2, 3, 2, 2

# Topological Sort




Valid BFS not valid DFS ?

What could stop a connected graph from running DFS

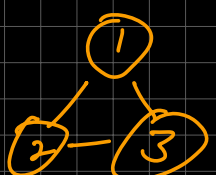



BFS = 1, 2, 3, 4, 5

DFS = 1, 2, 5, 4, 3

1 + 2

3)



$$\frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n\sqrt{5}}$$

```
public static int fib(int n)
  {
    double numerator;
    double k = 1 + Math.sqrt(5);
    double j = 1 - Math.sqrt(5);
     int two = 2;
     for(int i=0; i < n; i++){
          k *= k;
          j *= j;
          two *= two;
      }
    numerator = k - j;
    return (numerator/(two * Math.sqrt(5)));
  }
```

O(n)

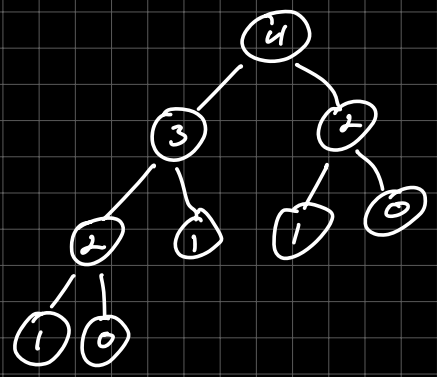
probably would have some rounding errors

```
  int [] memoize = new int[n+1]; → would be in wrapper
public static int fib(int n, int[] memoize)
{
      if(memoize[n] != null)
          return memoize[n]
      if(n == 0 || n == 1)
          return 1;
    memoize[n] = fib(n-1, memoize) + fib(n-2, memoize);
```

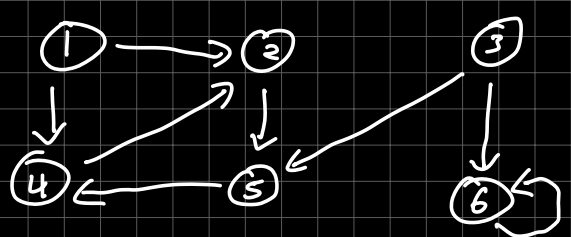
O(n)

return memoize[n];
}

algo O(n)




BFS:




either 1 or 3 are unreachable depending on breath origin

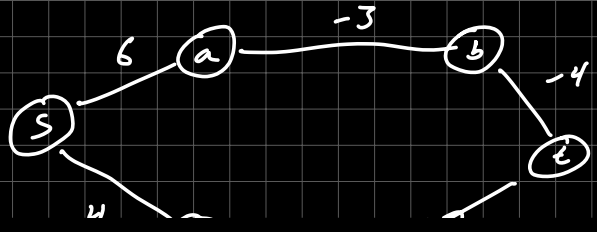
so we need to run 2 breadths at 1 & 3

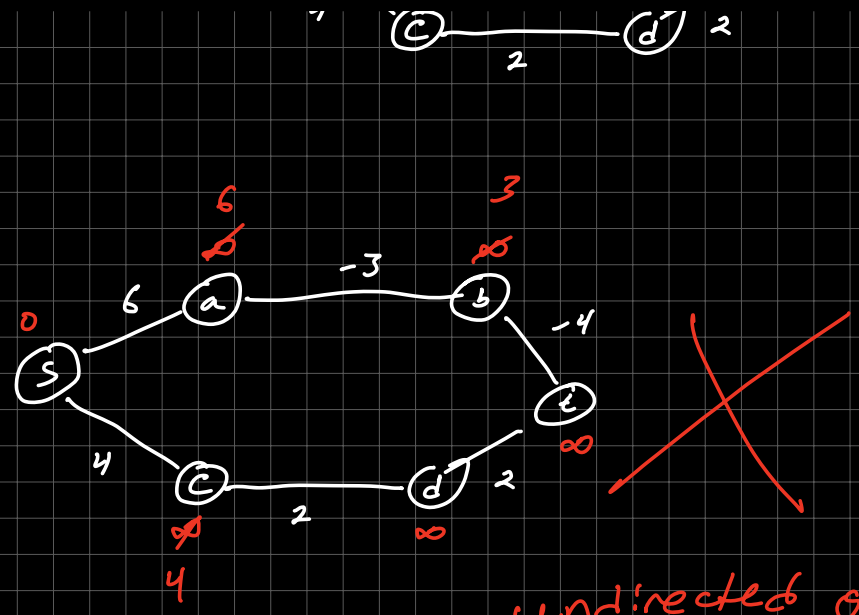
1: 1, 4, 2, 5

3: 3, 5, 6, 4, 2

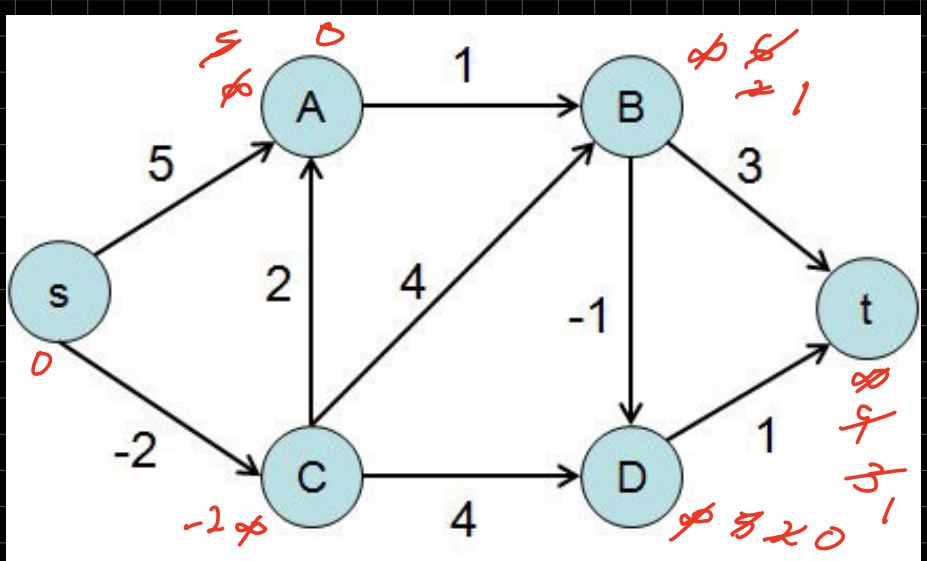
= 1, 4, 2, 5, 3, 6

> run one & share visited[]

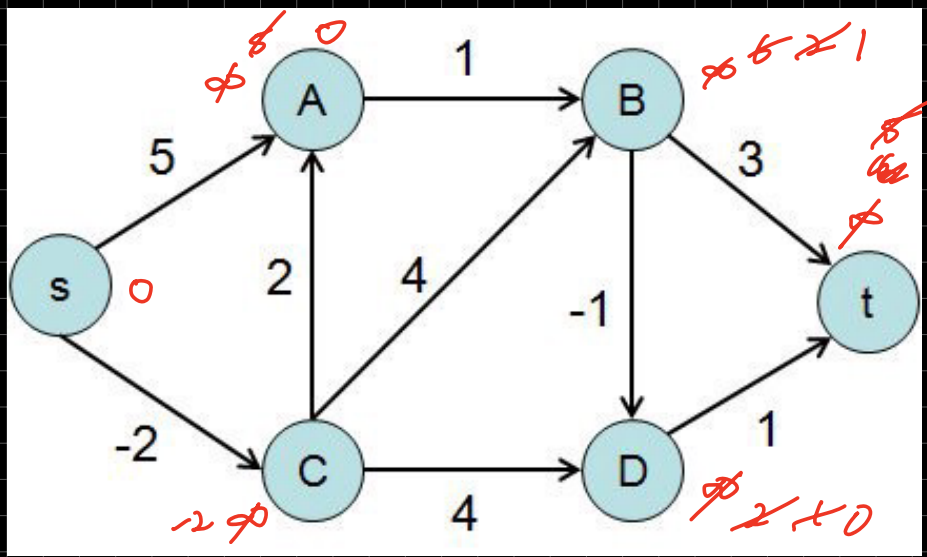





undirected graph w/ a negative edge w/ Bellman doesn't work




LBFORD

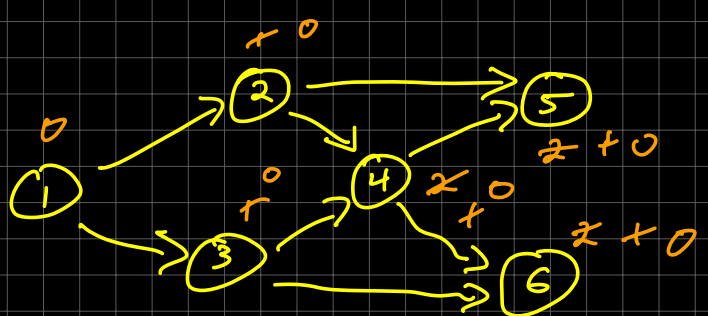



the modification would cause an infinite loop if there is a negative cycle

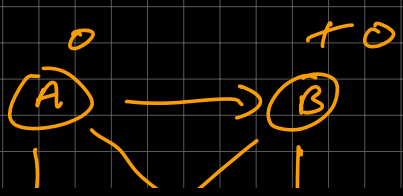
I would use a queue

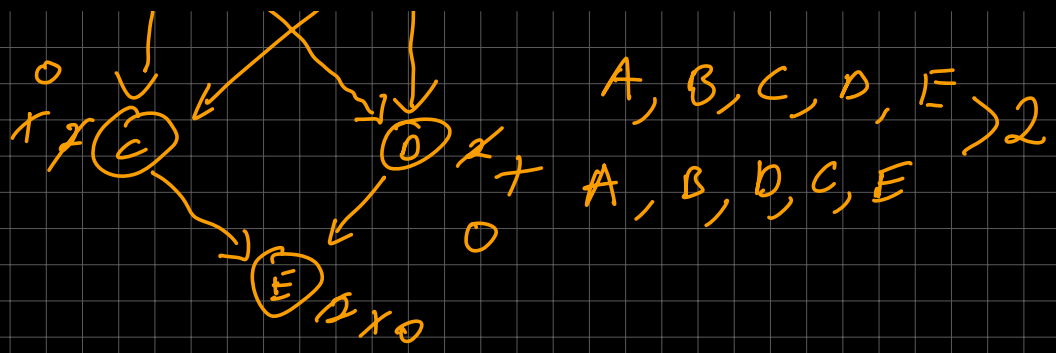
---

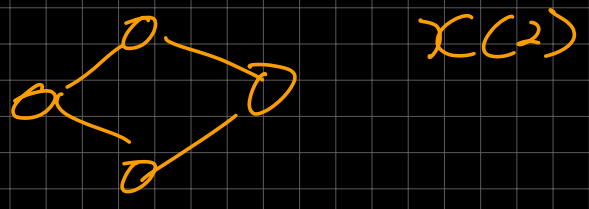
Topological Sort Practice

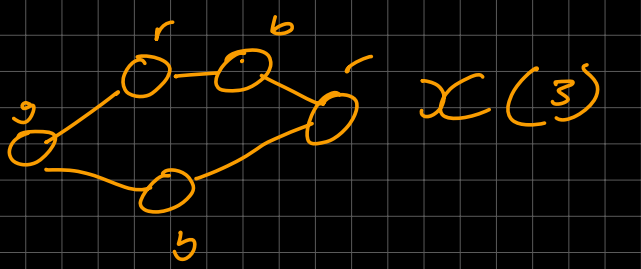



1, 3, 2, 4, 6, 5 6






what greedy algorithms have we seen?

Prim's — $O(|V| + |E|)$

Kruskal's — $O(|V| + |E|)$

a specific knapsack — recitation

a time slot problem — recitation

## guess runtimes

Bellman - $O(|V||E|)$ — uses priority queue

dijkstra's - $O(|E| + |V| \log |V|)$ w/ fib heap
$O((|E|+|V|)\log|V|)$ w/o fib heap
} can also use priority queue

topological - $O(|V| + |E|)$

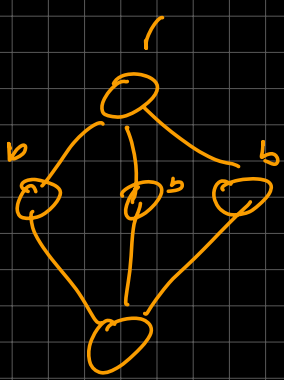
DFS & BFS - $O(|V|+|E|)$

Prim ~~$O((|V|+|E|)\log|V|)$~~

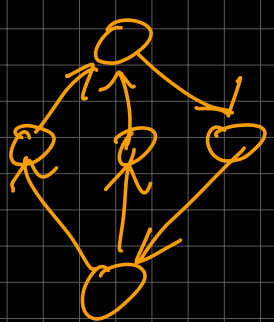
$O(|E| \log |V|)$

## Topological Sort

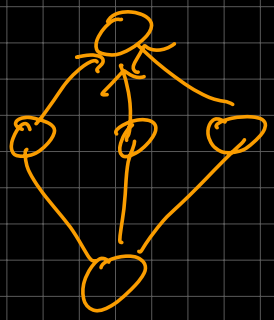
- do you use DFS to get indegree's for all nodes? store in array then add 0 degree nodes to queue




Eulerian path



Eulerian path

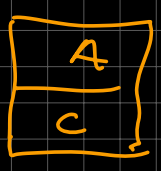


X

B, A, F, E, D, C

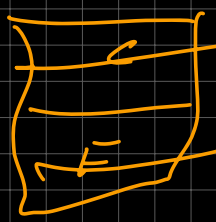
B,

stack




B, A,

stack




DFS = stack DS

BFS = Queue BQ

BQ DS

BQ DS

BFS = Queue  BQ

DFS = Stack  DS

Topological uses stack or Queue depending on BFS or DFS implementation