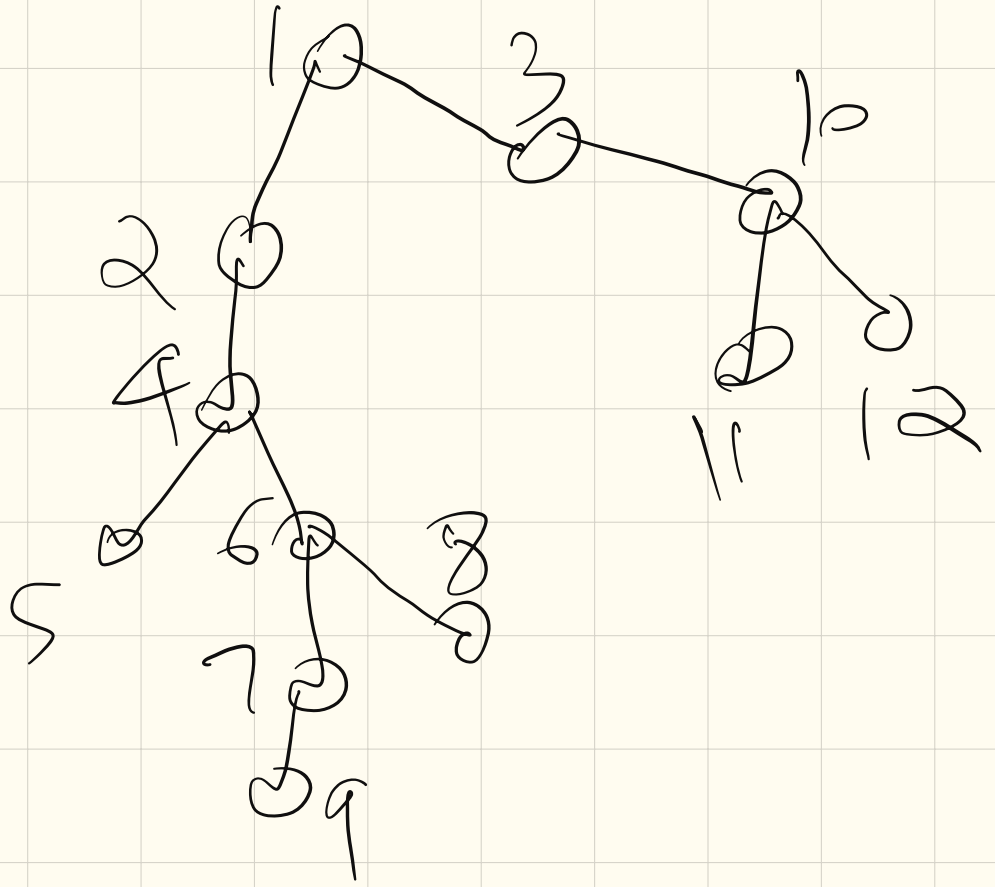
$$ecc(v) = \max_{u} \{d(u,v)\}$$

BFS(G, v)

- Find $\{ecc(v) : v \in V\}$
- Do BFS(G, v) $\forall v$
  - $|V|(|V|+|E|)$

Claim: We can find $\{ecc(v) : v \in V\}$ in $O(|V|)$ time for trees.

In particular, we can find $diam(G)$ in $O(|V|)$ time if $G \to$ tree.



$ecc(6)$

$= 6$

$6 - 12$,

$6 - 11$

paths

Observation:

Let s be an arbitrary vertex, v be a vertex

farthest from s.

Then v is the end-point of a diameter-length path.

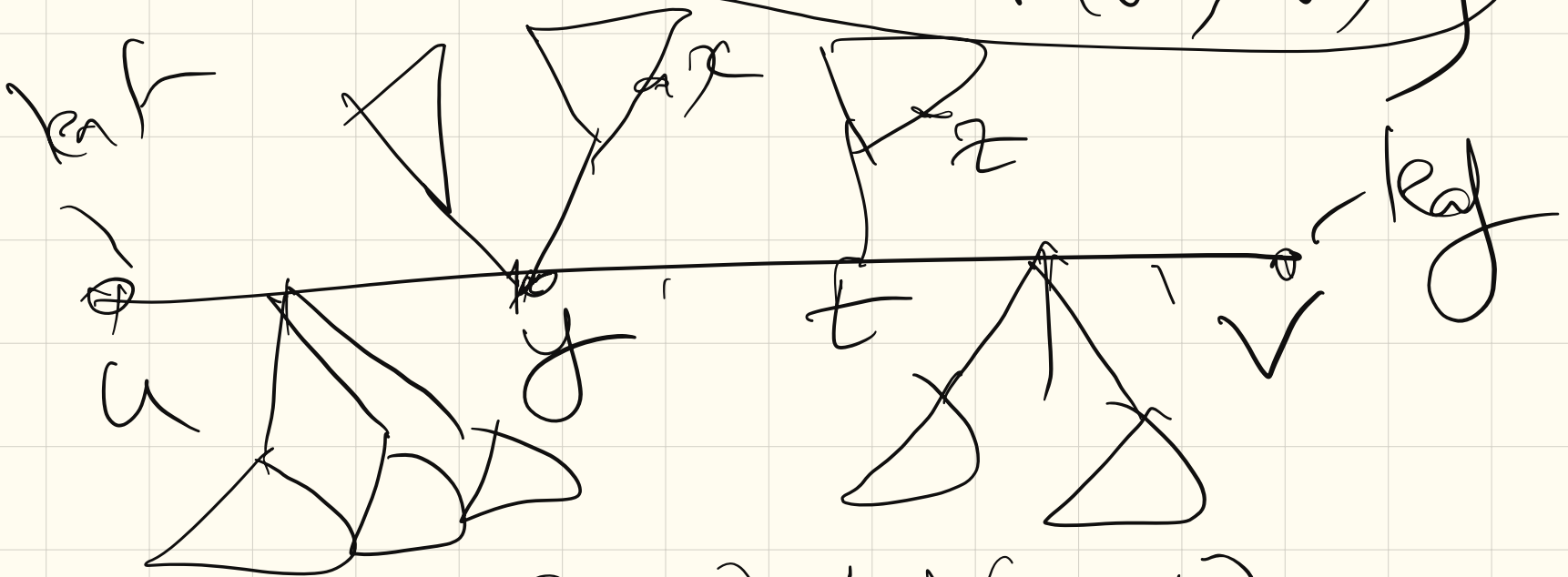
Algo to find diameter of a tree.

1. Pick an arbitrary vertex s.
2. Find v: $d(s,v)$ is max.
3. Find u: $d(v,u)$ is max.

Claim: $\forall w: ecc(w)$

$= \max\{d(\omega, x), d(\omega, y)\}$

$d(\omega, v)$



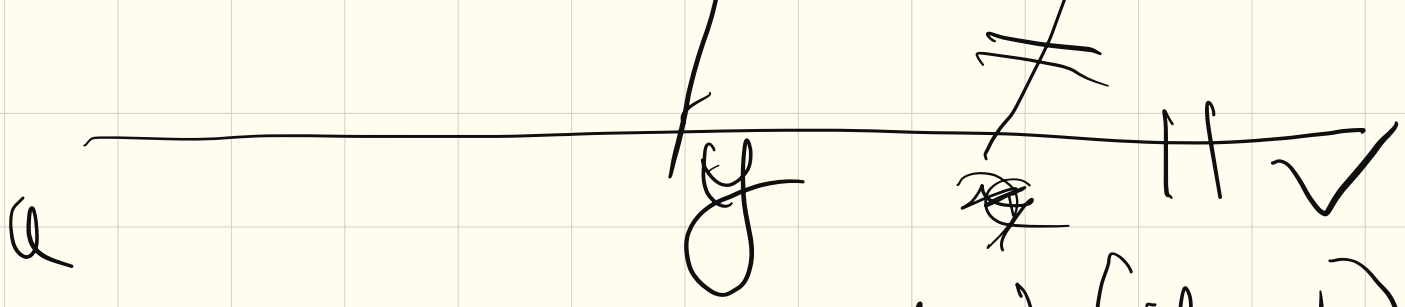
$d(x,z) \leq d(x,v)$

Suff: $d(t,z) \leq d(t,v)$

If $d(t,z) > d(t,v)$

$d(y,z) > d(y,v)$

I^x I^z

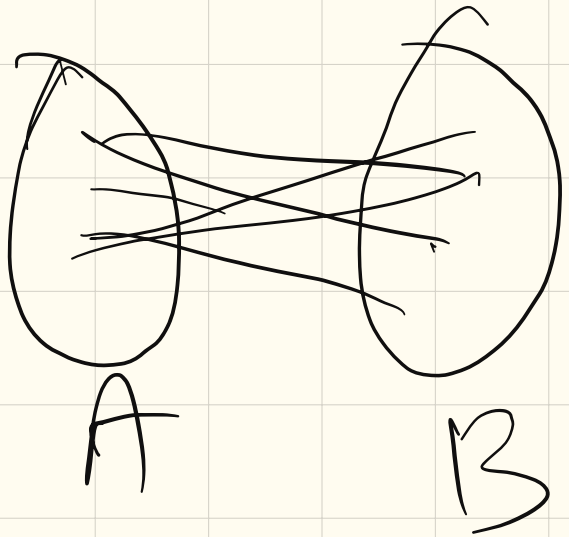


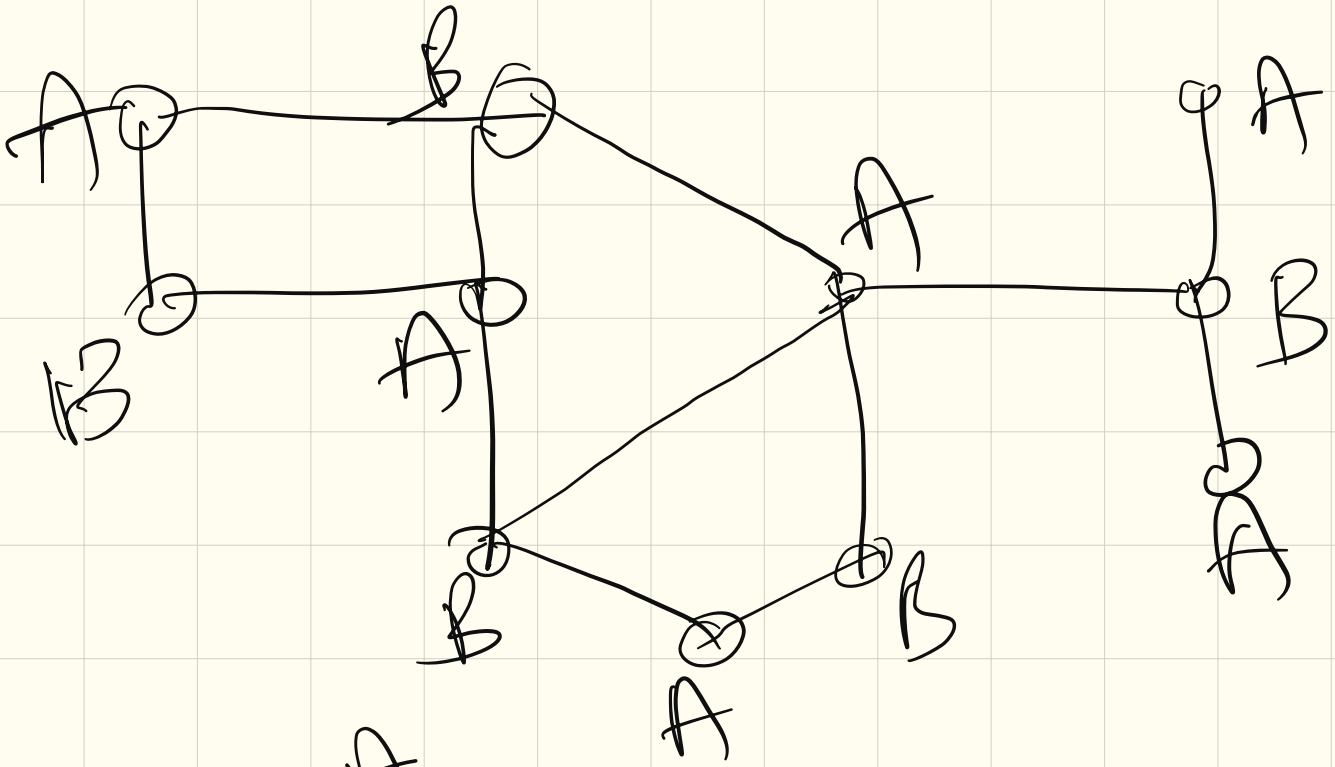
$d[x,y) \leq d(y,v)$

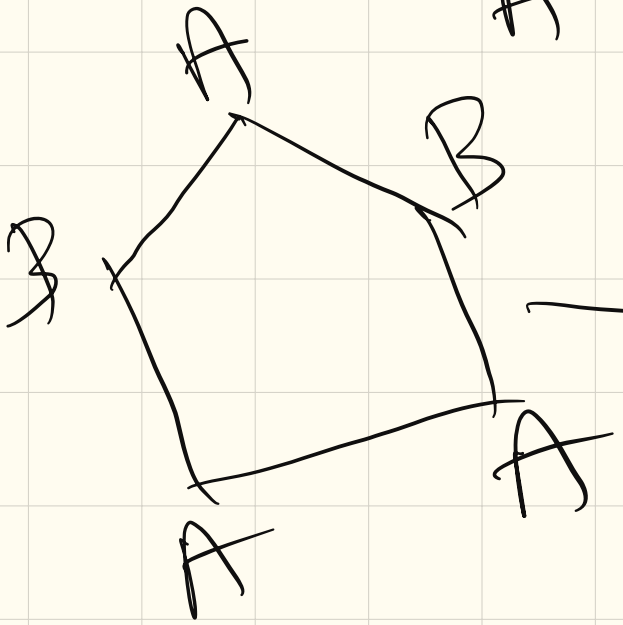
$d(x,y) \leq d(u,y)$

z is nearest from x.

Bipartite graphs

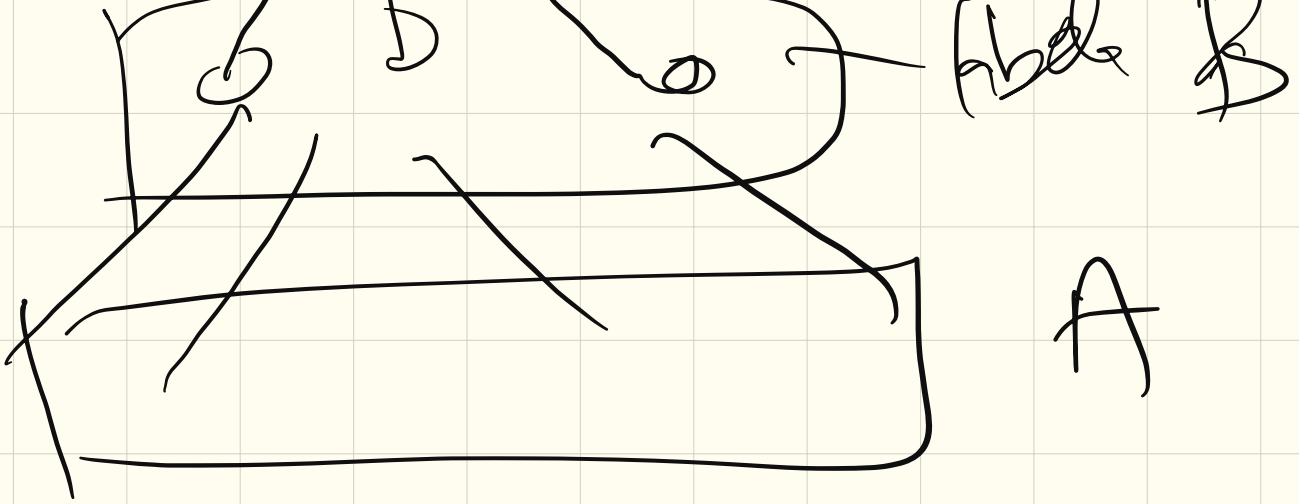






→ not bipartite

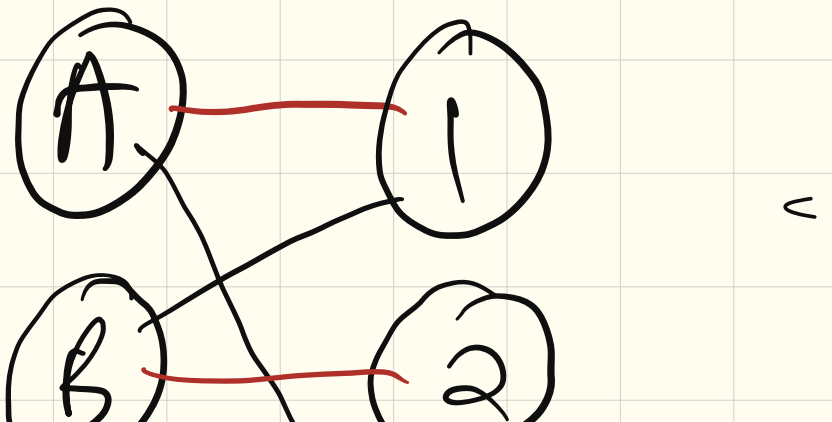
→ label as A

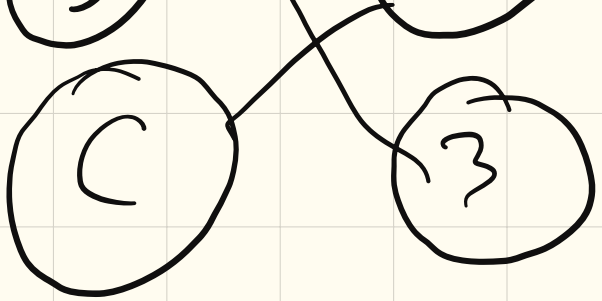


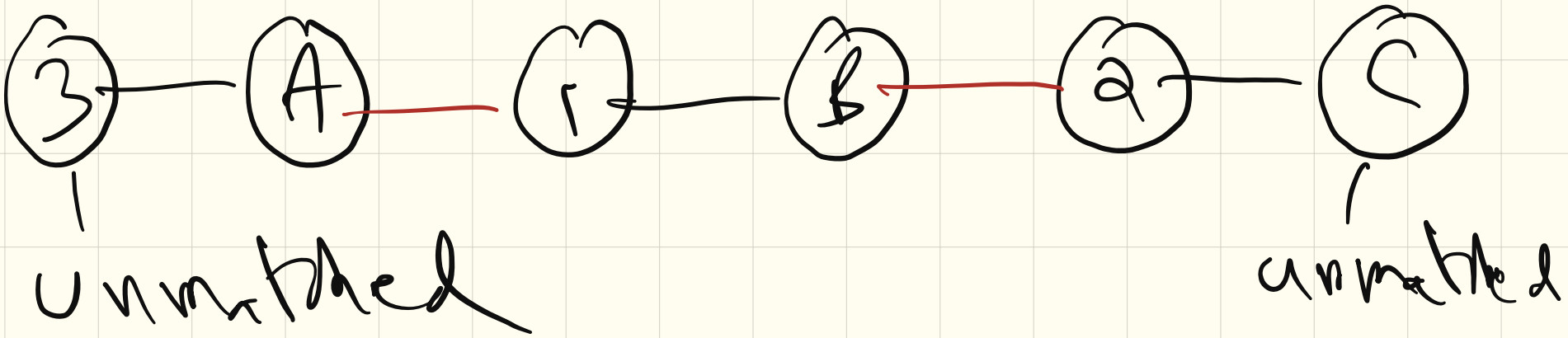
G is bipartite

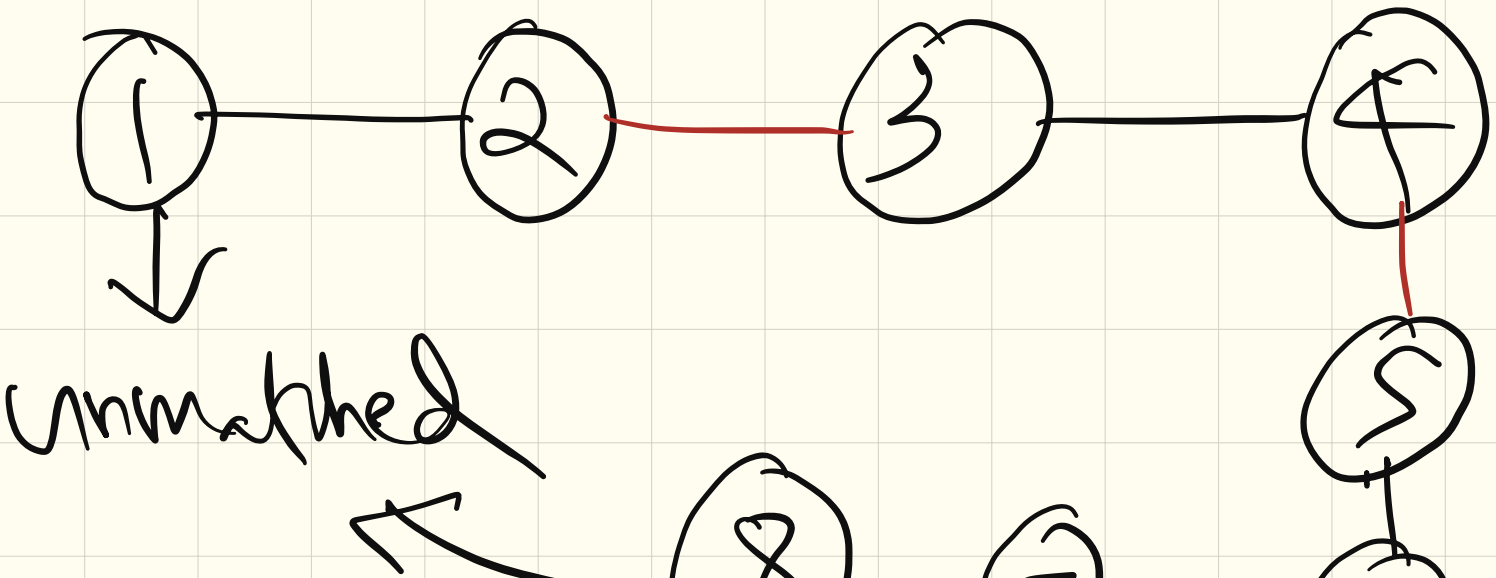
$\Leftrightarrow$

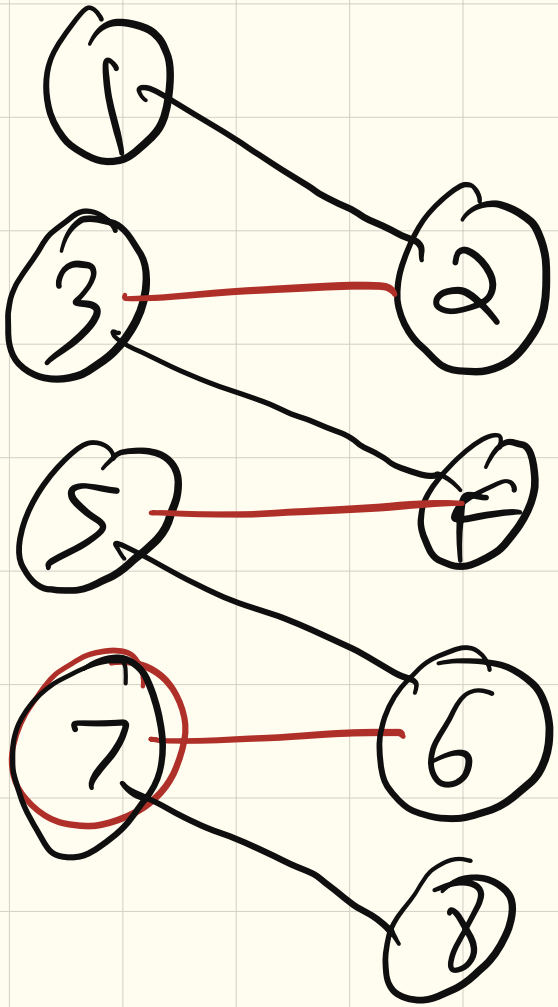
G has no odd cycles.




C
3

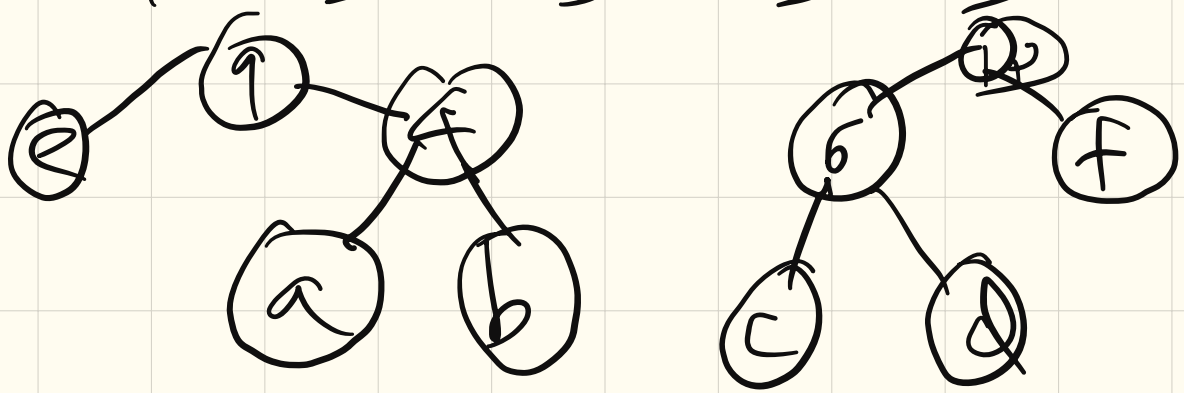

3
A
1
B
2
C
unmatched
unmatched


1
2
3
4
5
unmatched



a b c d e f g

1 3 3 3 5 6 16



F/p: String S, list L:
collection of
pairs of letters.

O/p: Longest subsequence of S
avoiding pairs in L appearing
consecutively.

$f(i)$ = Longest subsequence of $S[1] \dots S[i]$ ending at i

$g(i,j)$ = Longest P subseq. of $S[i] \dots S[j]$
$i \le j$

$$f(i) = \max_{j<i} f(j) + 1$$

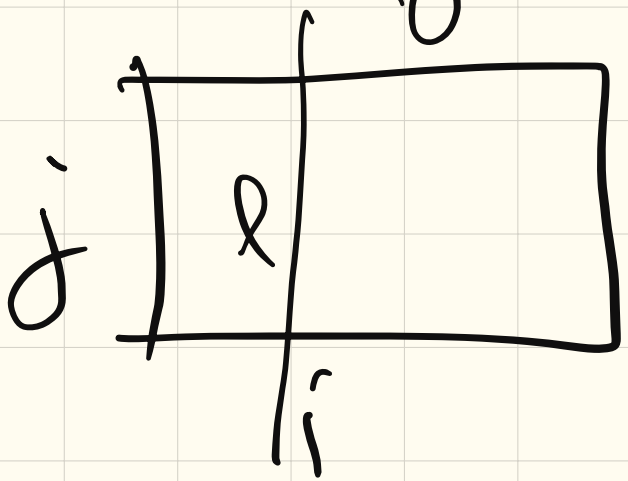
$S[j]S[i] \notin L$

$S[j]S[i] \notin L$

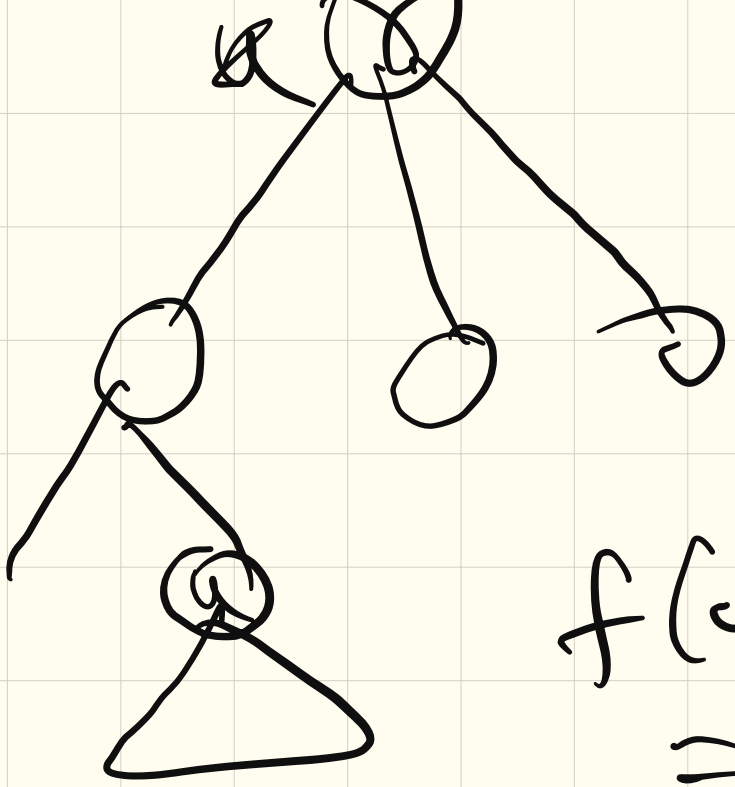
Time: $O(n^2)$

Q3: $f(i,j)$ = Optimal path for a $i \times j$ cake

$$f(i,j) = \text{Max} \{ P_{i,j}, \max_{1 \le l \le i-1} (f(l,j) + f(i-l,j)), \max_{1 \le k \le j-1} (f(i,k) + f(i,j-k)) \}$$



Time: $O(mn(m+n))$



$f(u)$
= Max weight matching for T_u

$f(u, o)$
= Max weight matching for T_u, where