CS6130 : Advanced Graph Algorithms

Matchings in general graphs

- Algorithm

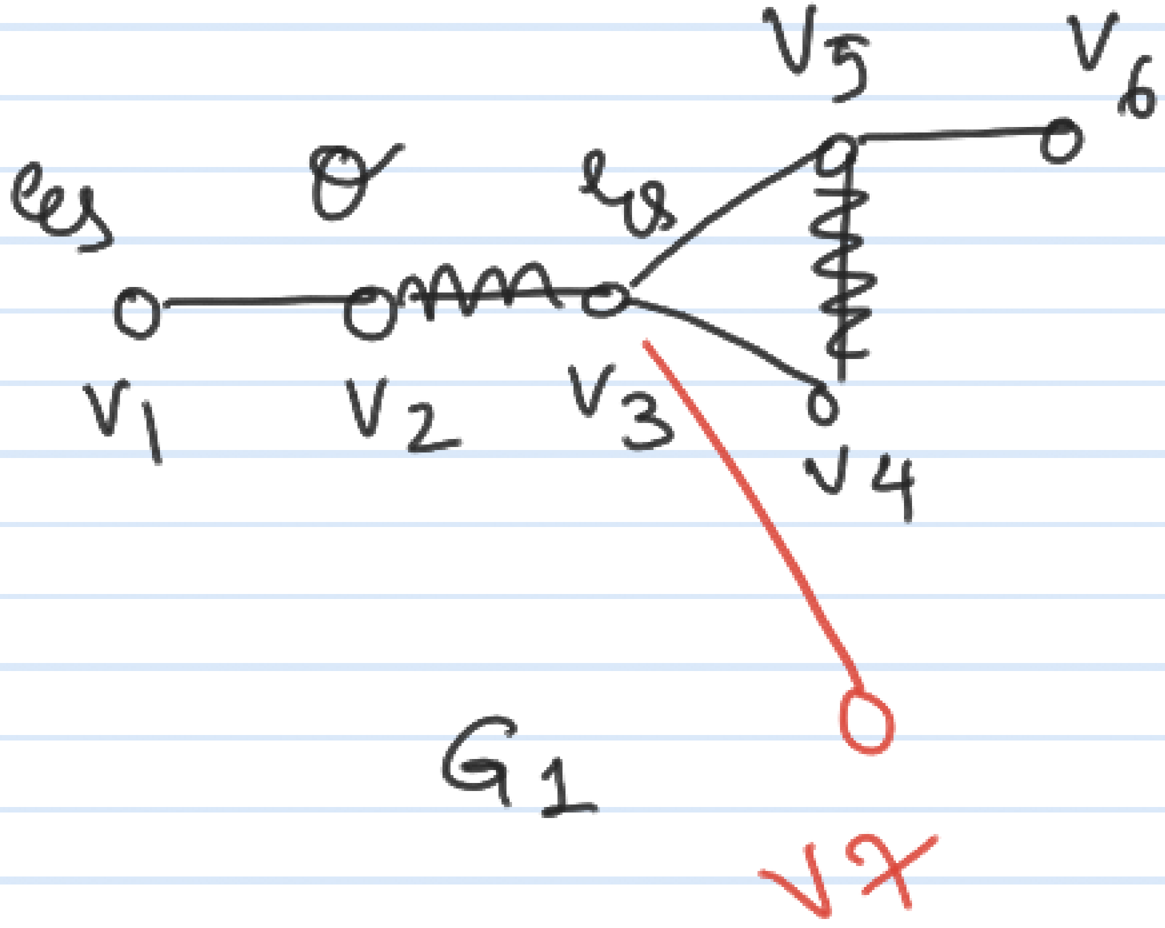
- Certificate of Optimality

- Tutte's theorem

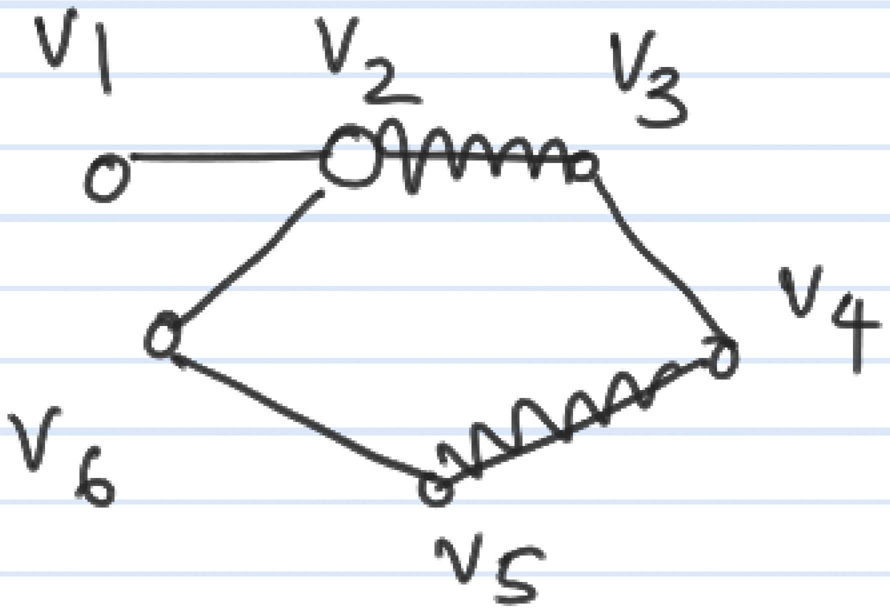
- Properties invariant of max matching

Berge's theorem and augmenting paths

How to compute aug. path efficiently?



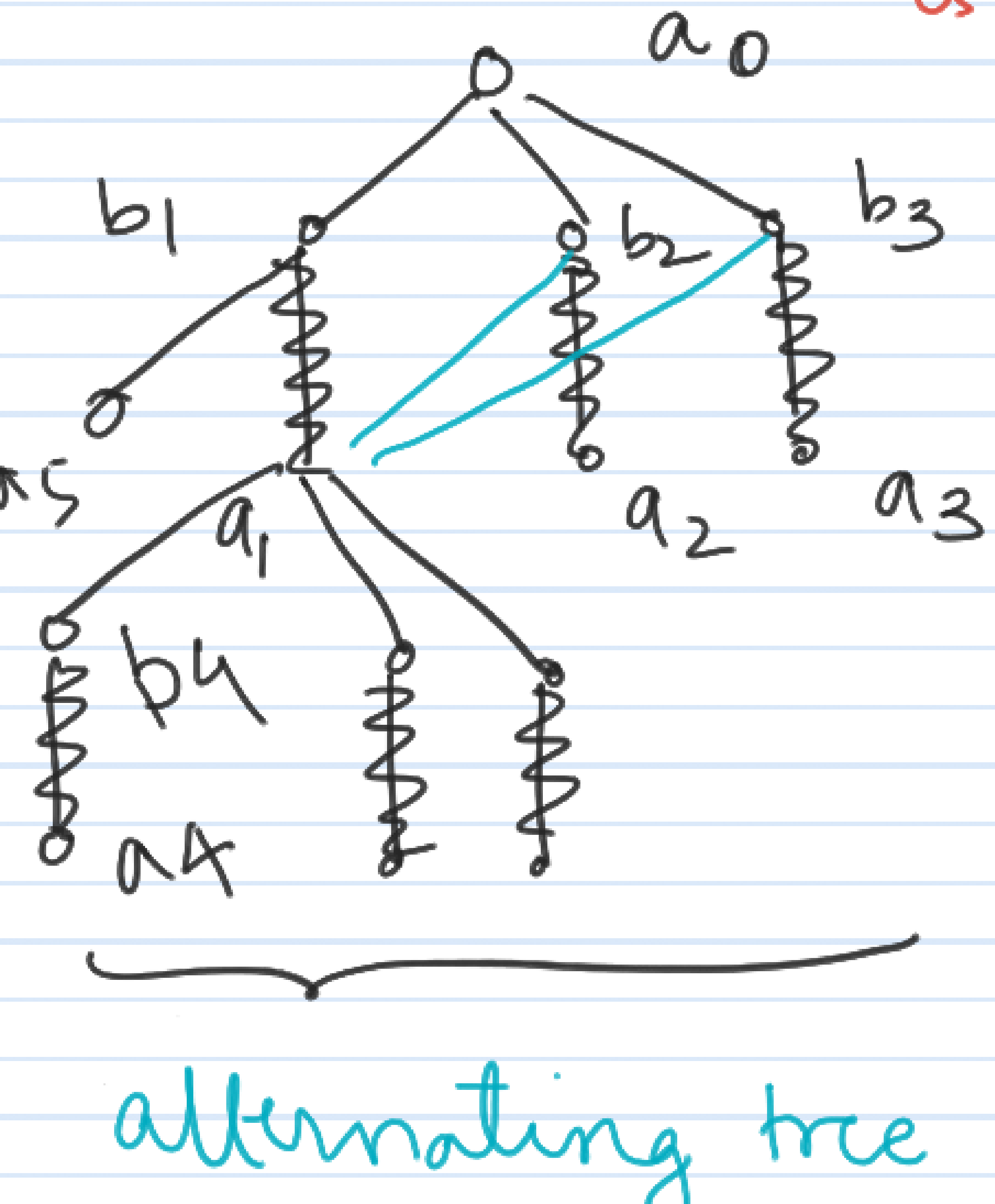
G1



G2

Berge's theorem and augmenting paths

Revisiting the bipartite case in this light



yes

- where do the non tree edges belong?

no

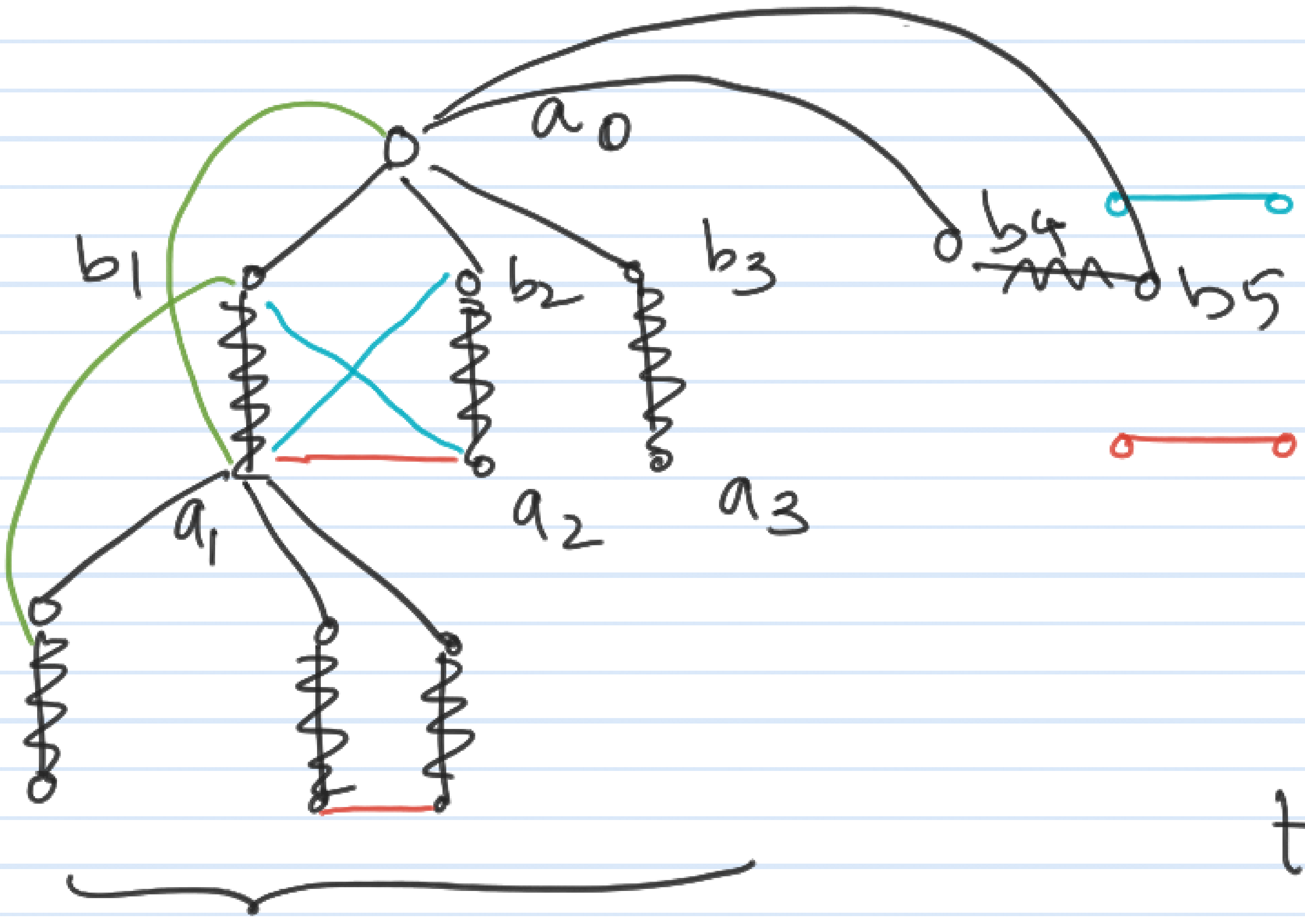
yes

- how do they affect our exploration?

no

Berge's theorem and augmenting paths

Back to general graphs



do not alter labels

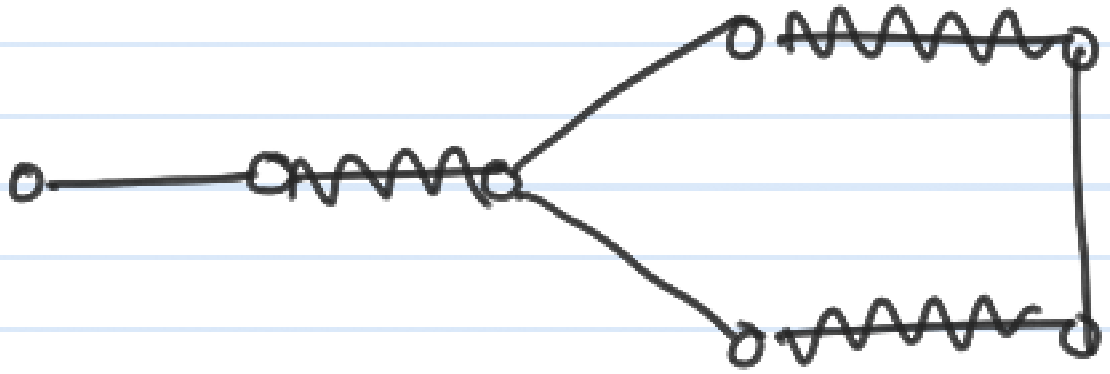
flip the labels

not-to be-explored
$\Downarrow$
to be explored.

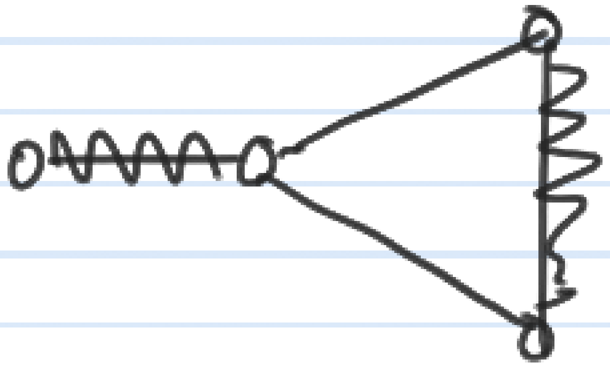
are there any other "type" of edges?

Edmond's idea of a blossom

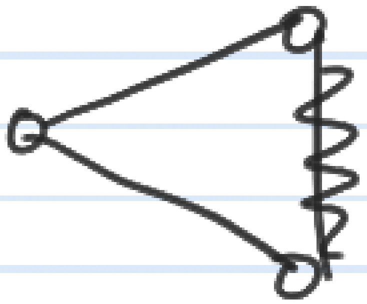
- odd cycle with specific properties



C_1



C_2



C_3

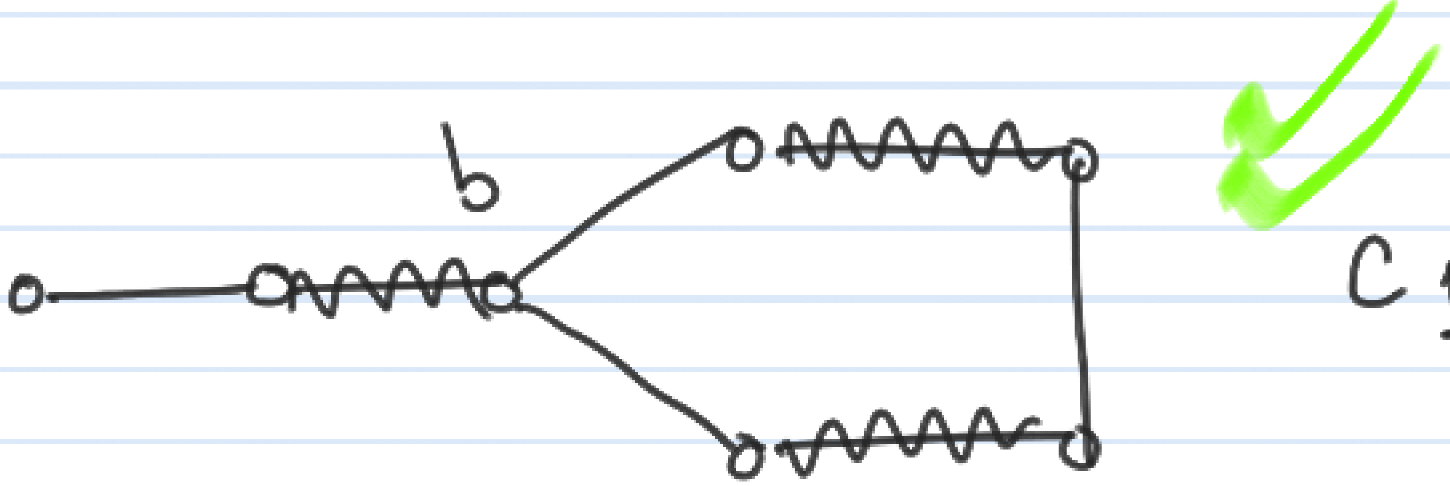
JACK EDMONDS

Seminal paper
"Paths Trees and Flowers"

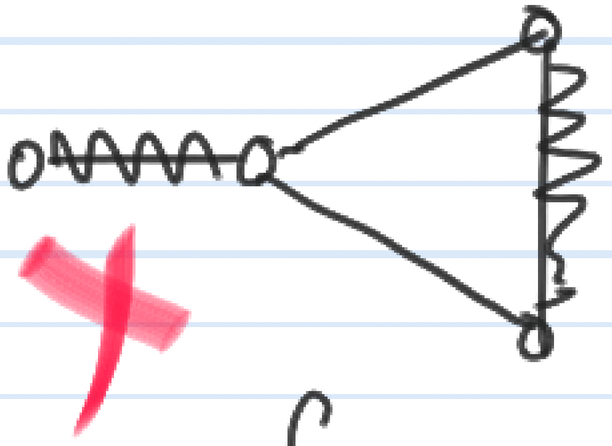
- recognized Ptime as notion of efficiency

Edmond's idea of a blossom

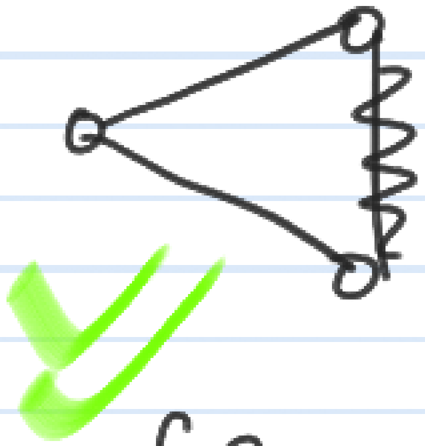
- odd cycle with specific properties



C_1



C_2



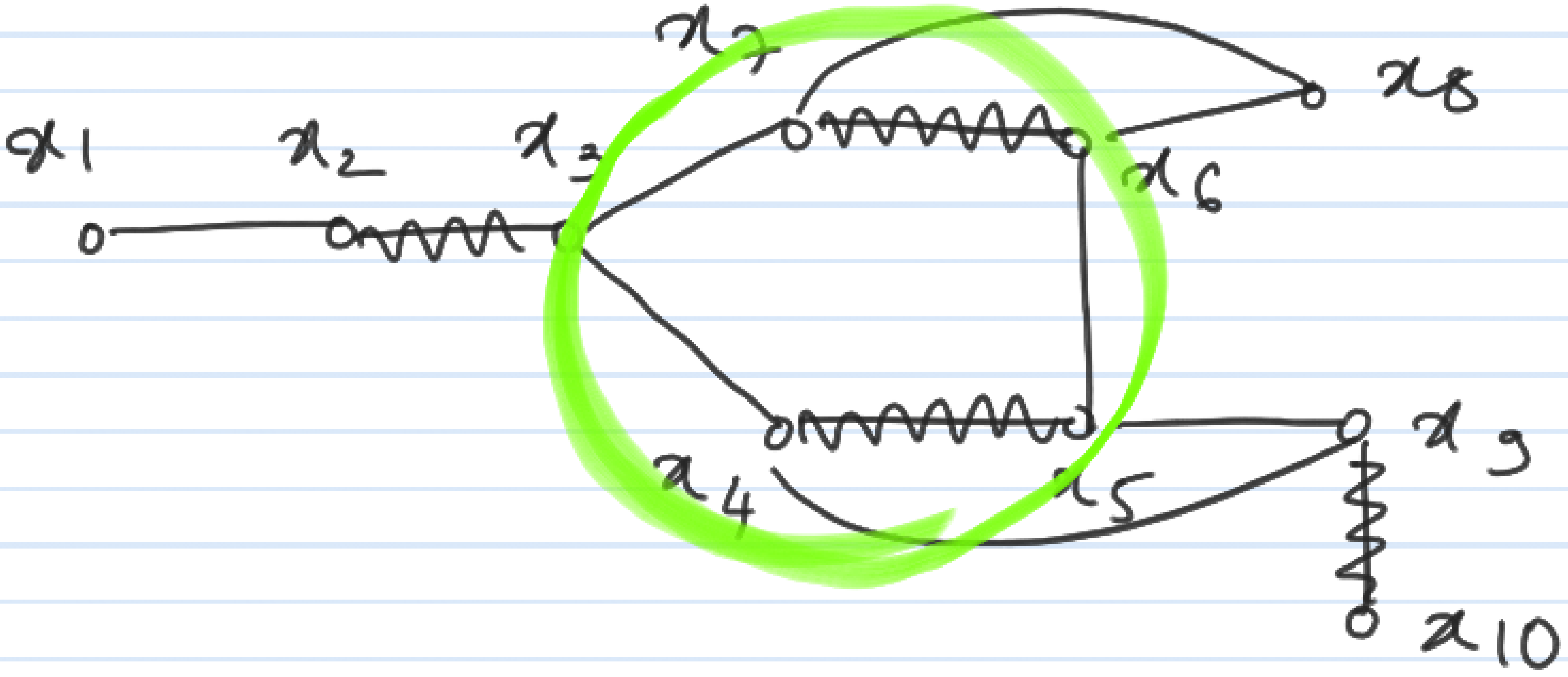
C_3

- odd cycle which is maximally matched inside C
- unique vertex not matched inside the cycle is "even"

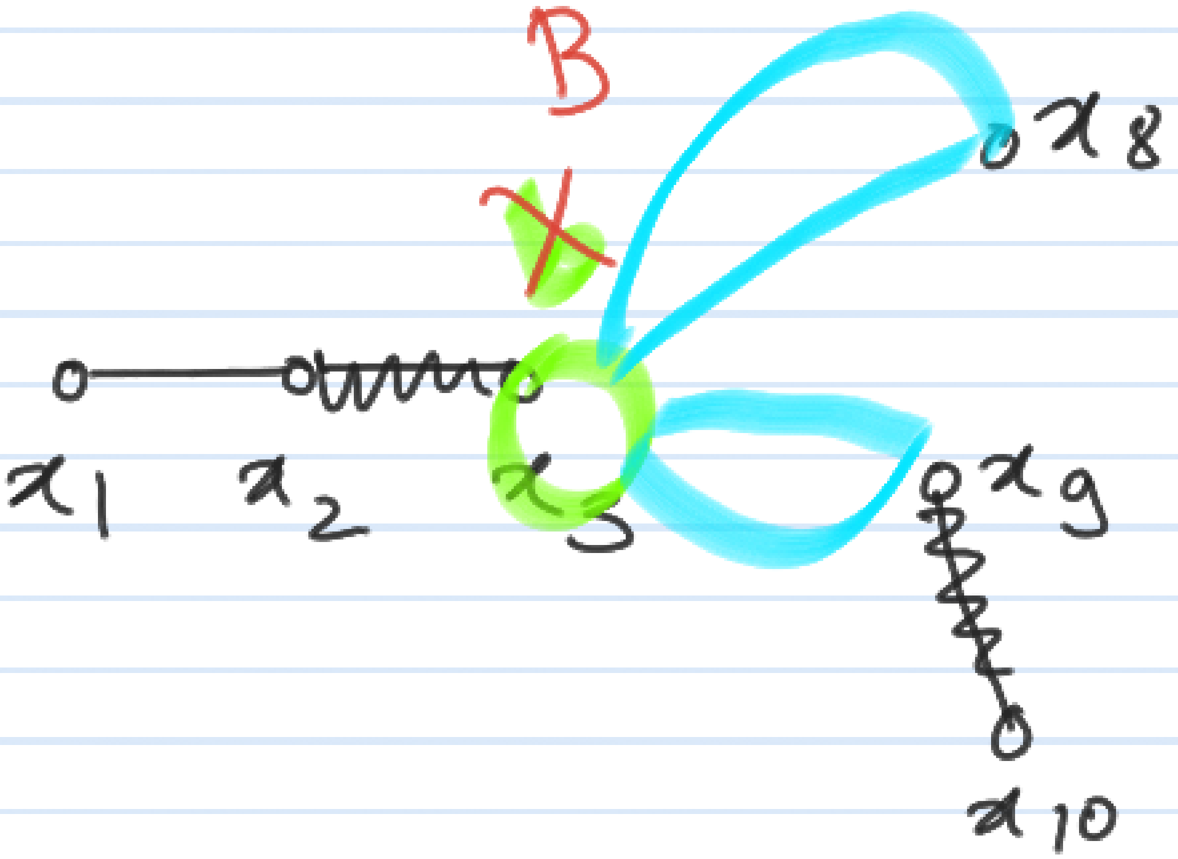
Blossom : Definition and examples

- An odd cycle C which is maximally matched inside the cycle
- unique vertex of C which is unmatched in C is either unmatched in M or is reachable via an even length alternating path starting at a free vertex.

Shrinking a blossom

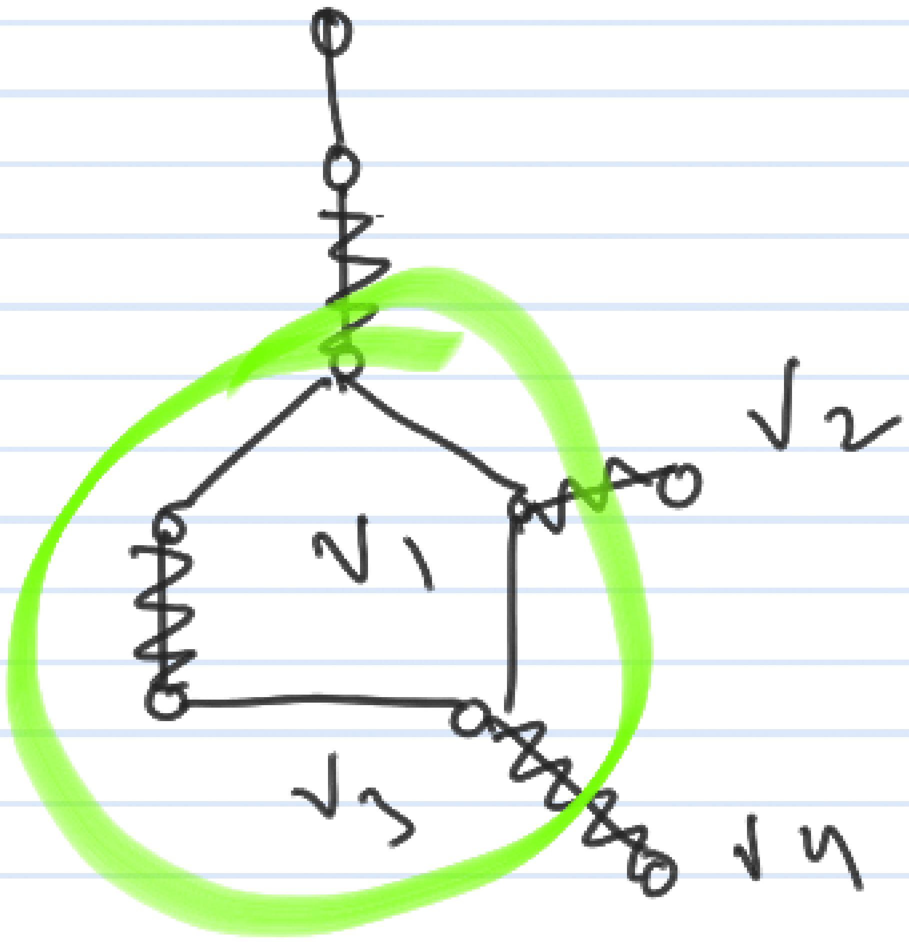


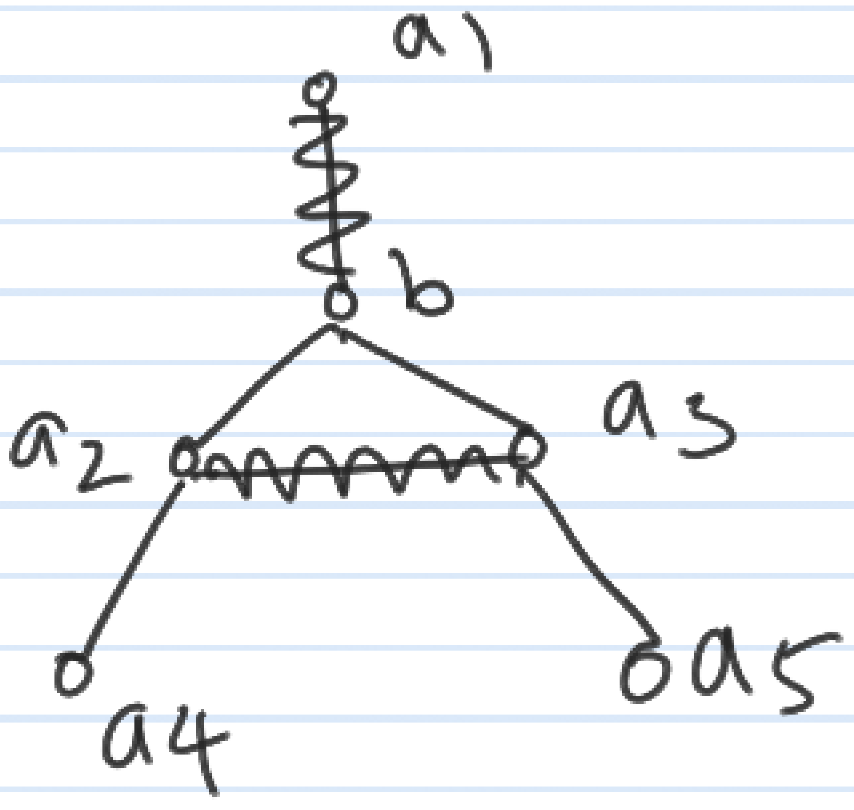
G , M

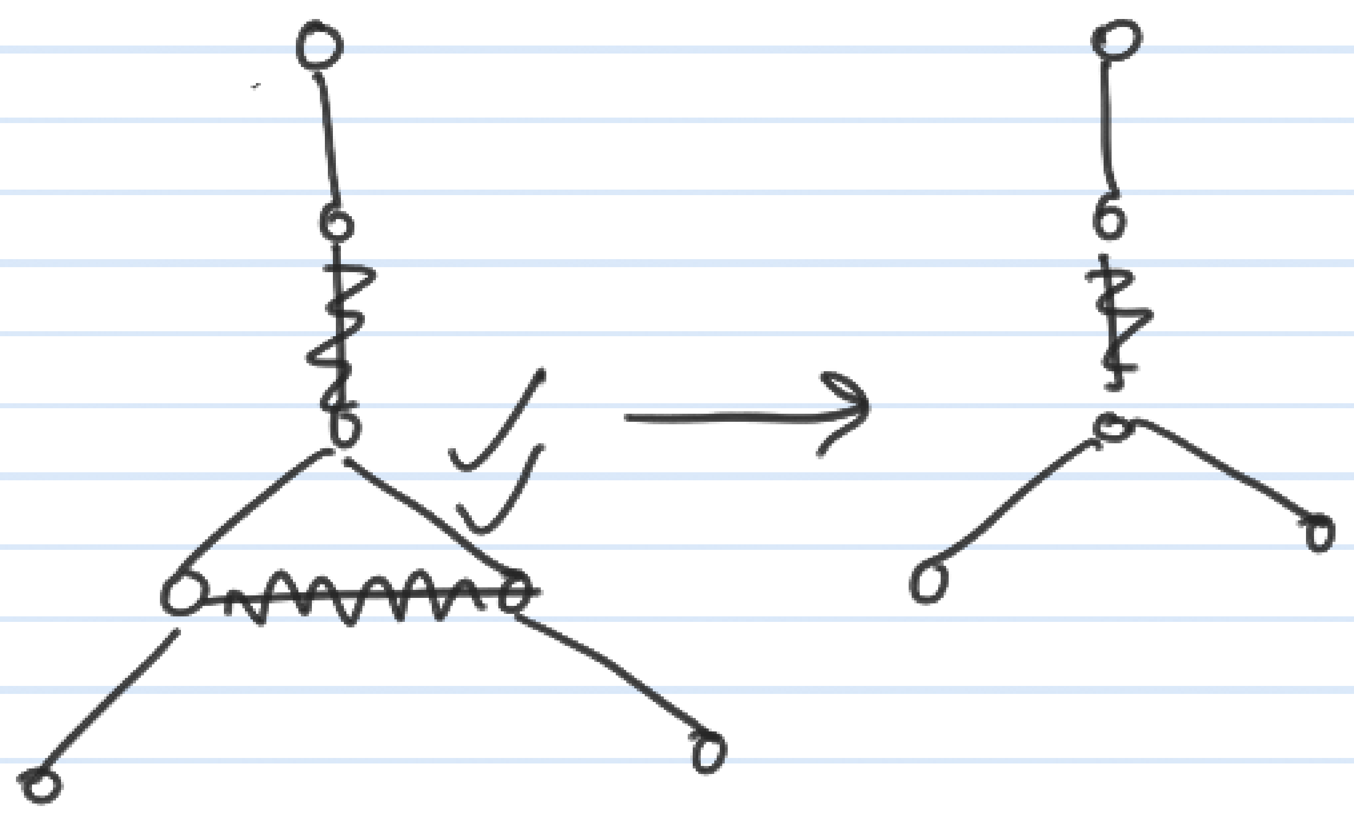


$G|B$, $M|B$

Define $G|B$ } how are they useful?
$M|B$

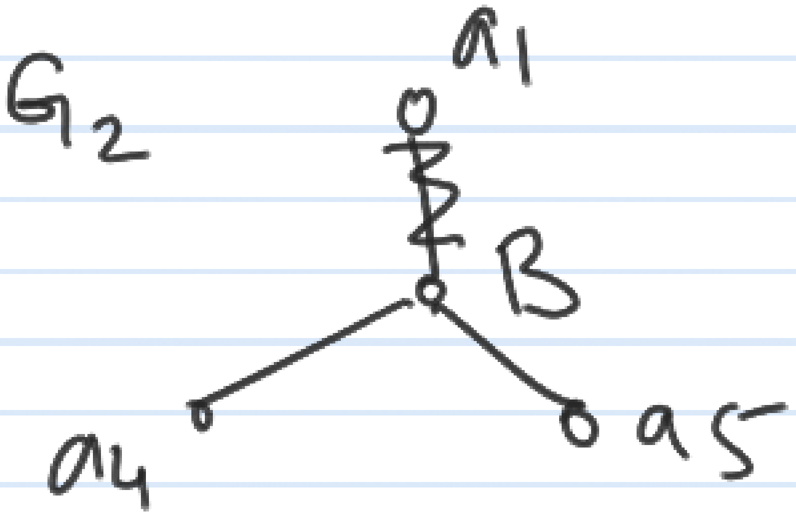






G1

G2



G3

G, M   B

G|B   M|B

shrinking cycles in each of the cases.

The shrunk graph and shrunk matching

G $\longrightarrow$ G/B

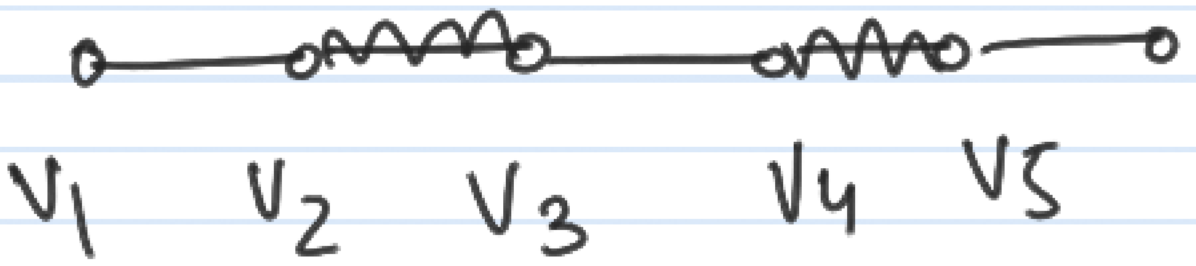
M, B $\quad$ M/B

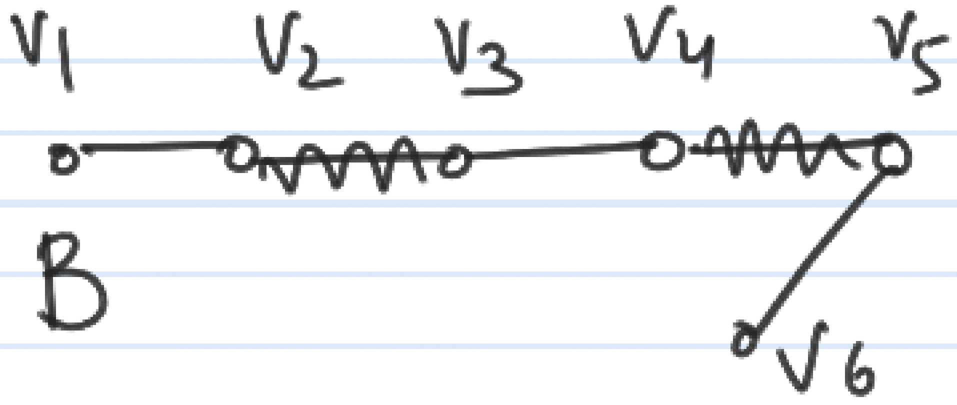
any matching

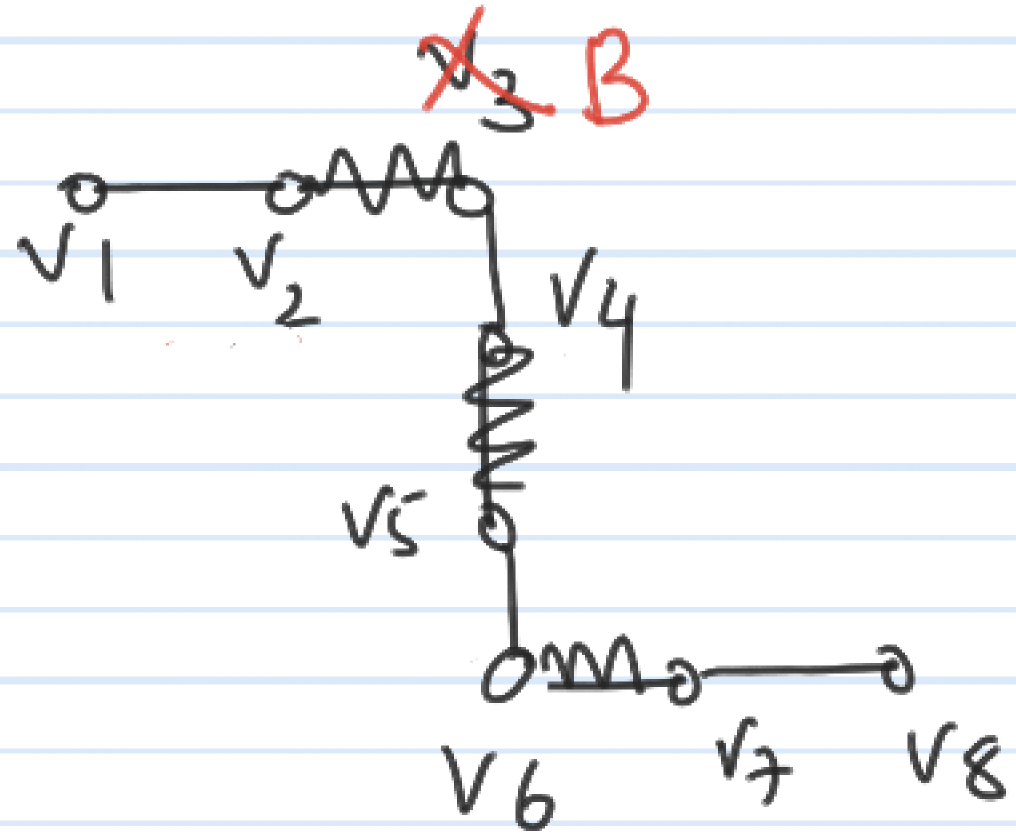
a blossom w.r.t. M in G

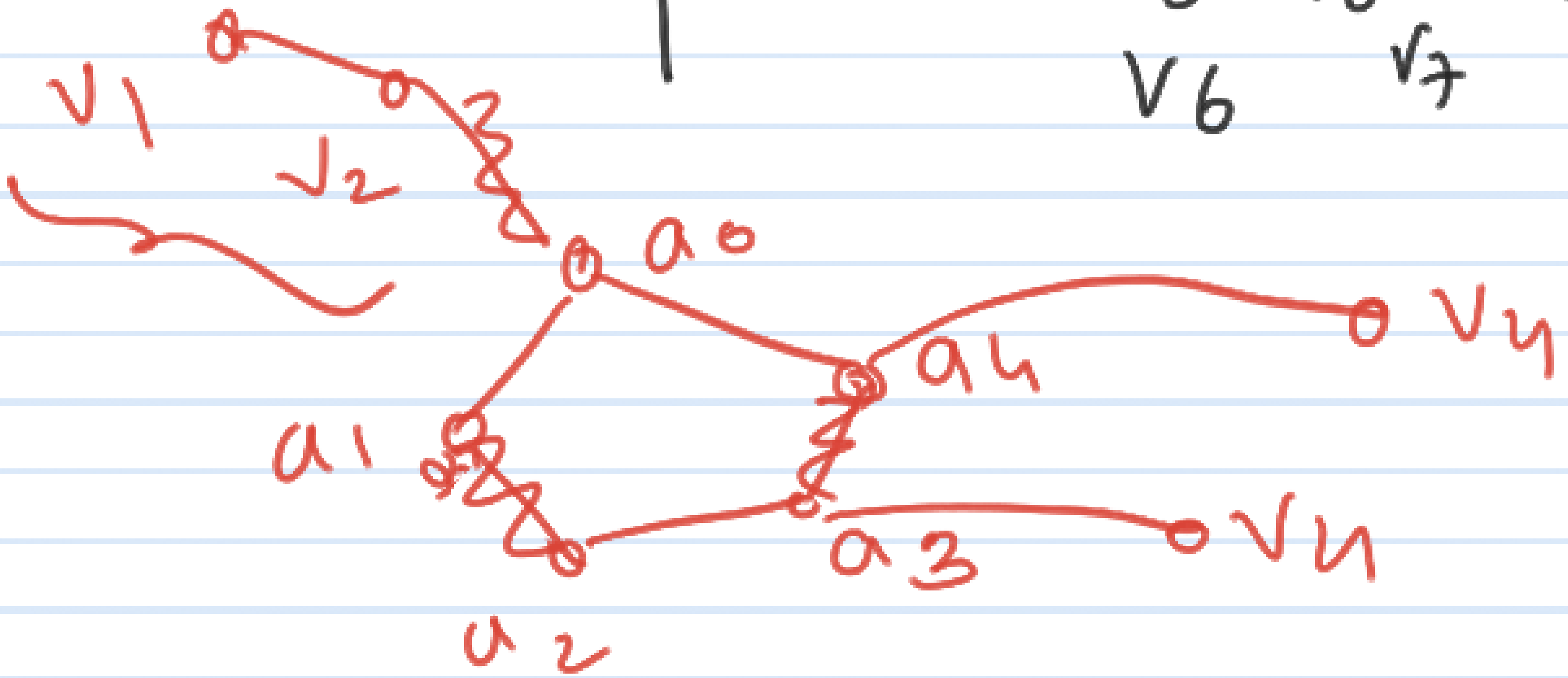
Claim: $\exists$ an aug path w.r.t. M in G iff $\exists$ an aug path w.r.t M/B in G/B

Assume that $\exists$ a path P w.r.t $M|B$ in $G|B$









Cases:

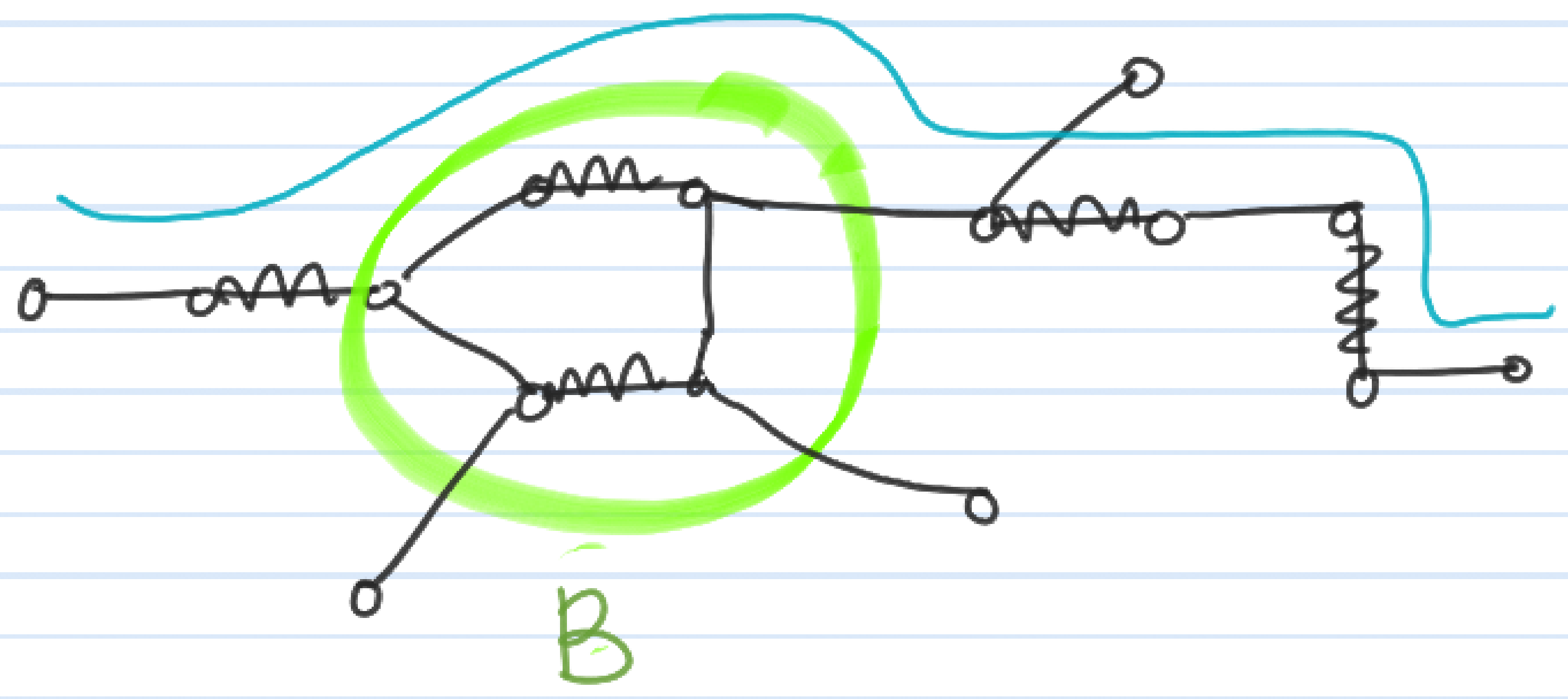
(1) P does NOT contain B

(2) P contains B

↳ and hence B was unmatched in M

↳ and hence B is matched in M

Assume that $\exists$ a path P w.r.t M/B in G/B



By definition of blossom every vertex in B is reachable via an even length alternating path start. at a free vertex in M

Cases:

(1) P does NOT contain B

(2) P contains B

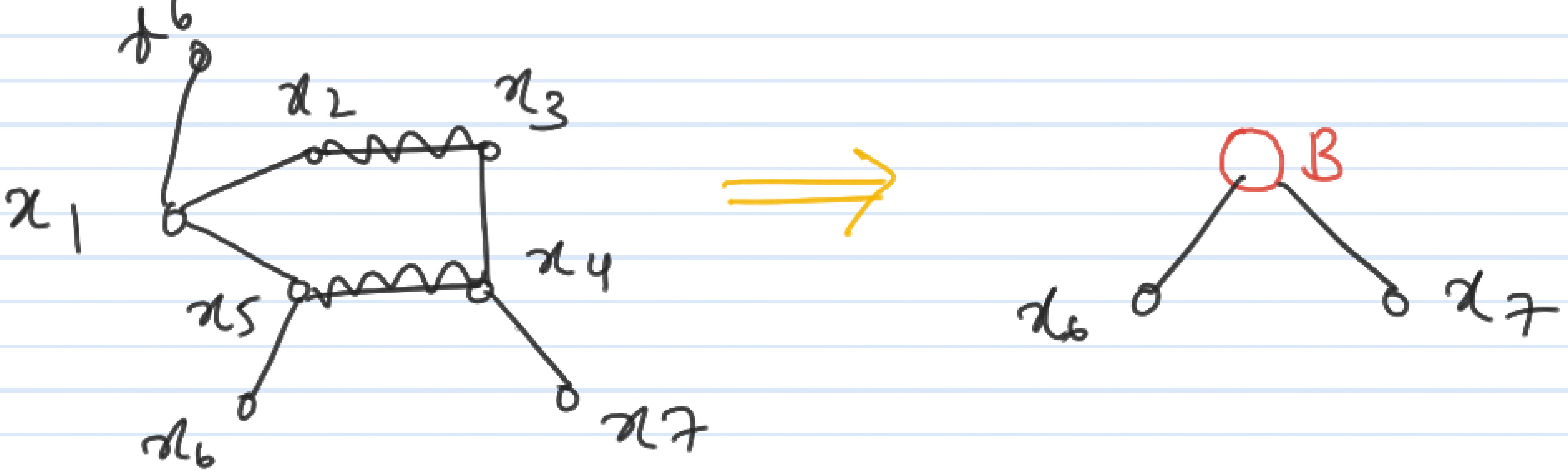
b and hence B was unmatched in M

b and hence B is matched in M

Other Direction : Assume $\exists$ a path P in G w.r.t M

Note : all paths are not preserved.

, So proof is needed.



- does P contain b?
- is b matched in M or not?

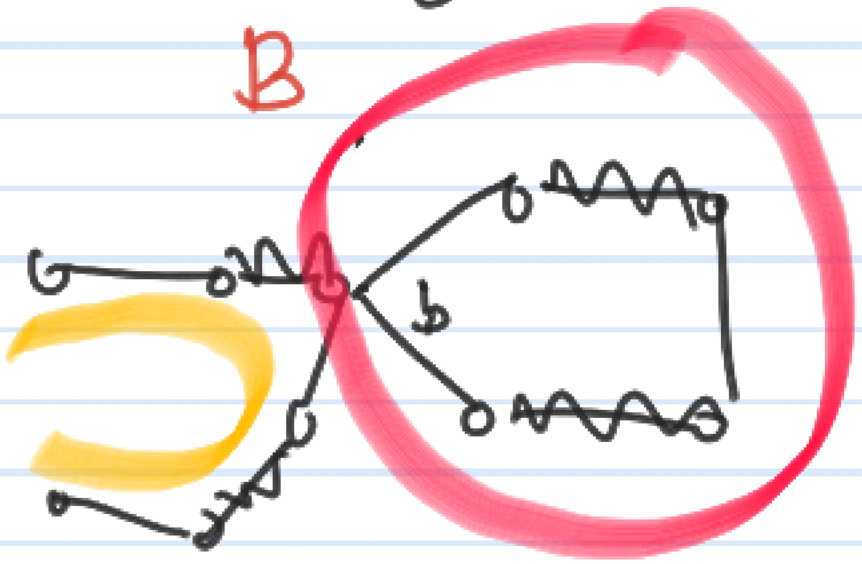
Other Direction : Assume $\exists$ a path P in G w.r.t M

Note : all paths are not preserved.
So proof is needed.

- P contains b

P contains b only

B



P contains b and some more

- P does NOT contain b

– P must have odd number of edges from the cycle

High level algorithm

G : a general graph ; M : any matching

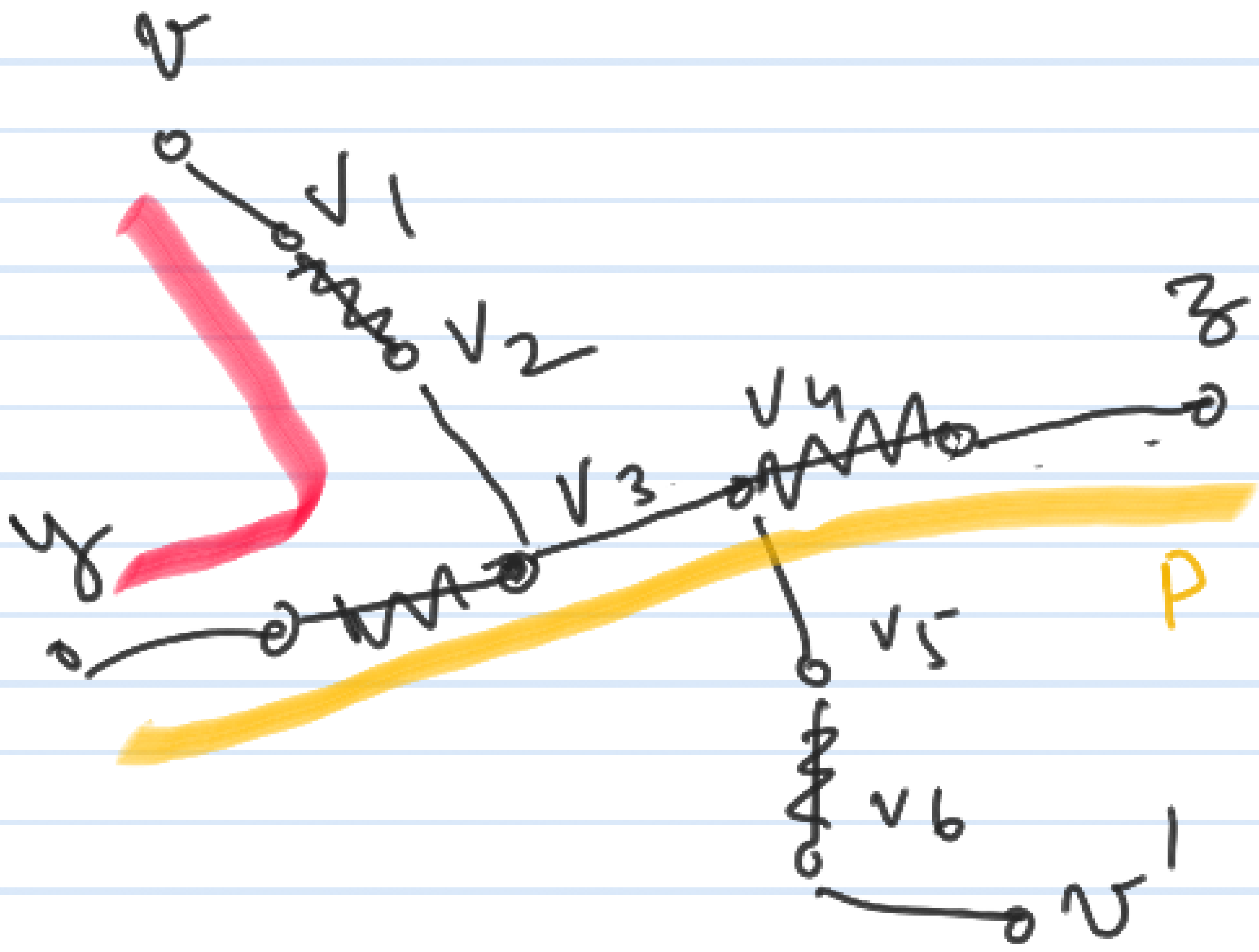
- pick an unmatched vertex v } n times
- explore v, build an alternating tree } $O(m)$
  - ↳ in this process find a blossom or aug. path
- if aug path found, modify M } $O(mn)$
- if blossom B found, shrink $\underbrace{G \to G/B \text{ and } M \to M/B}_{O(mn)}$
- if none found, $\underbrace{\text{discard } v \text{ and pick another vertex.}}_{\text{needs justification}}$

Overall time : $O(m\, n^2)$

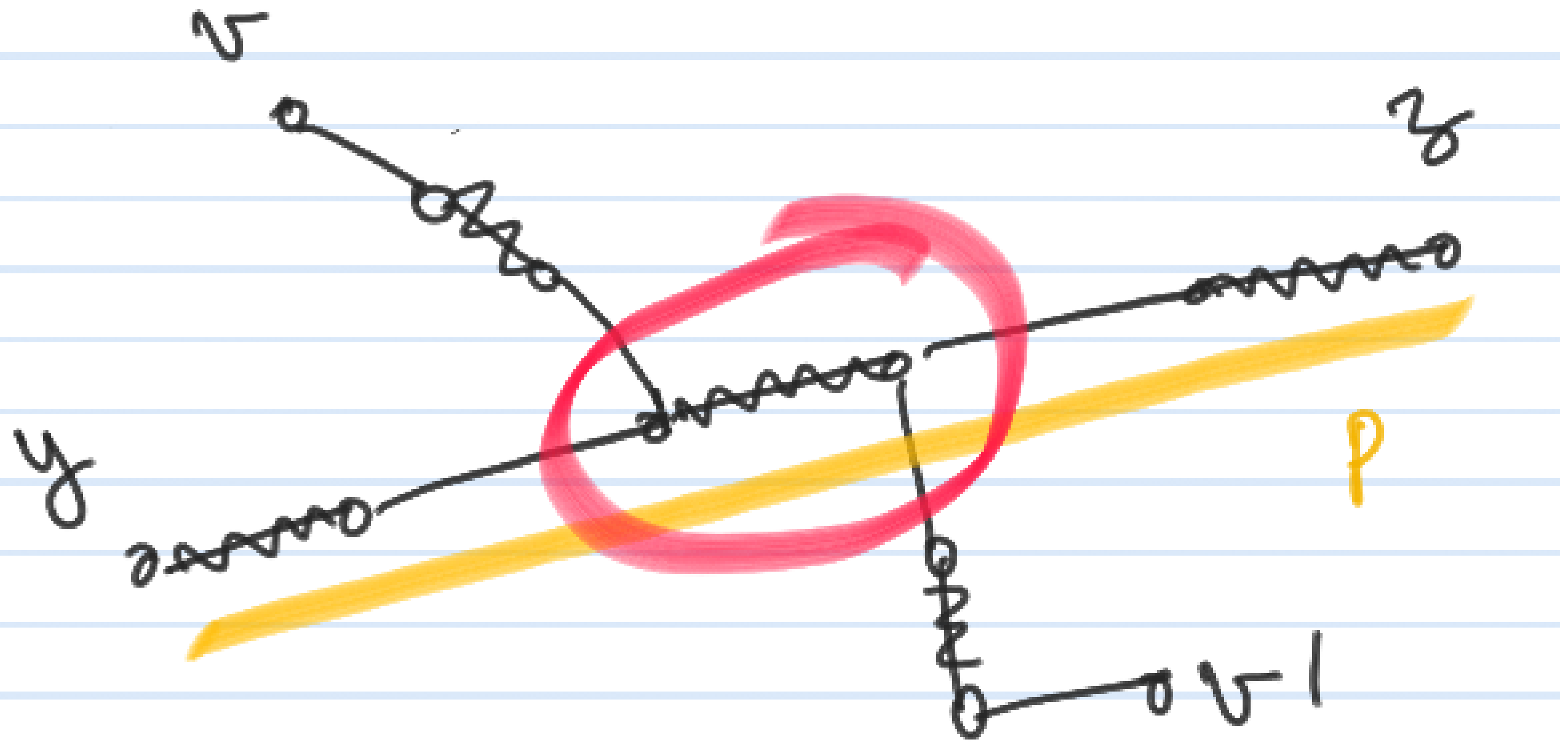
A vertex needs to be explored at most once

Claim: If $\nexists$ is no aug path w.r.t M starting at v, then $\nexists$ no aug path w.r.t

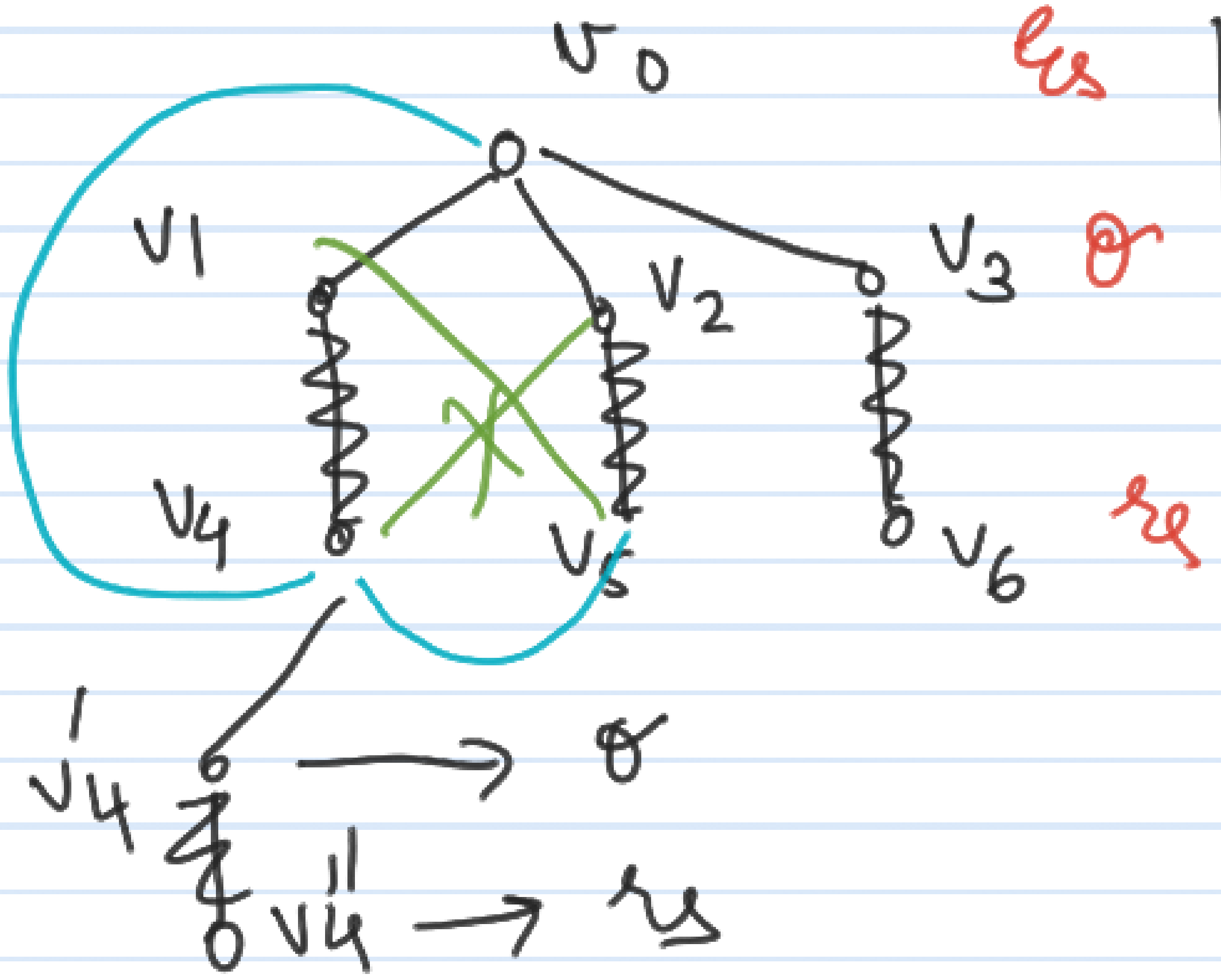
$$M \oplus P$$



$\Leftarrow$



Data structures and detecting blossoms



Vertex specific DS

pred[v] → pred vertex in tree

label[v] → odd / even / NULL

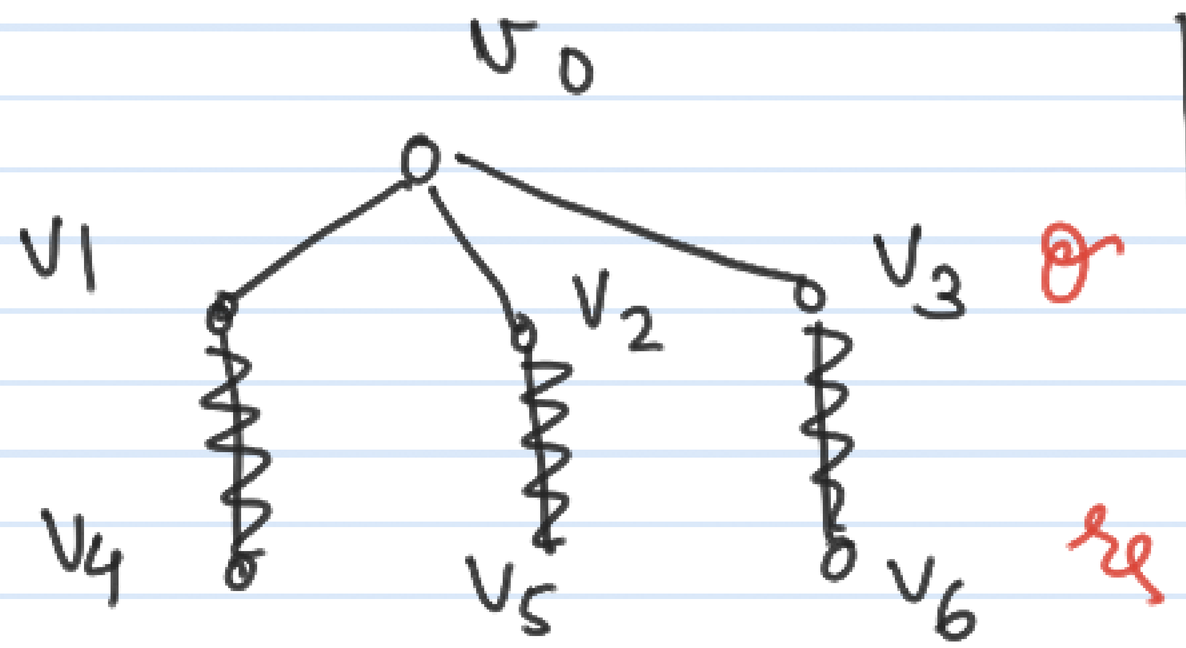
~~B[v]~~ →

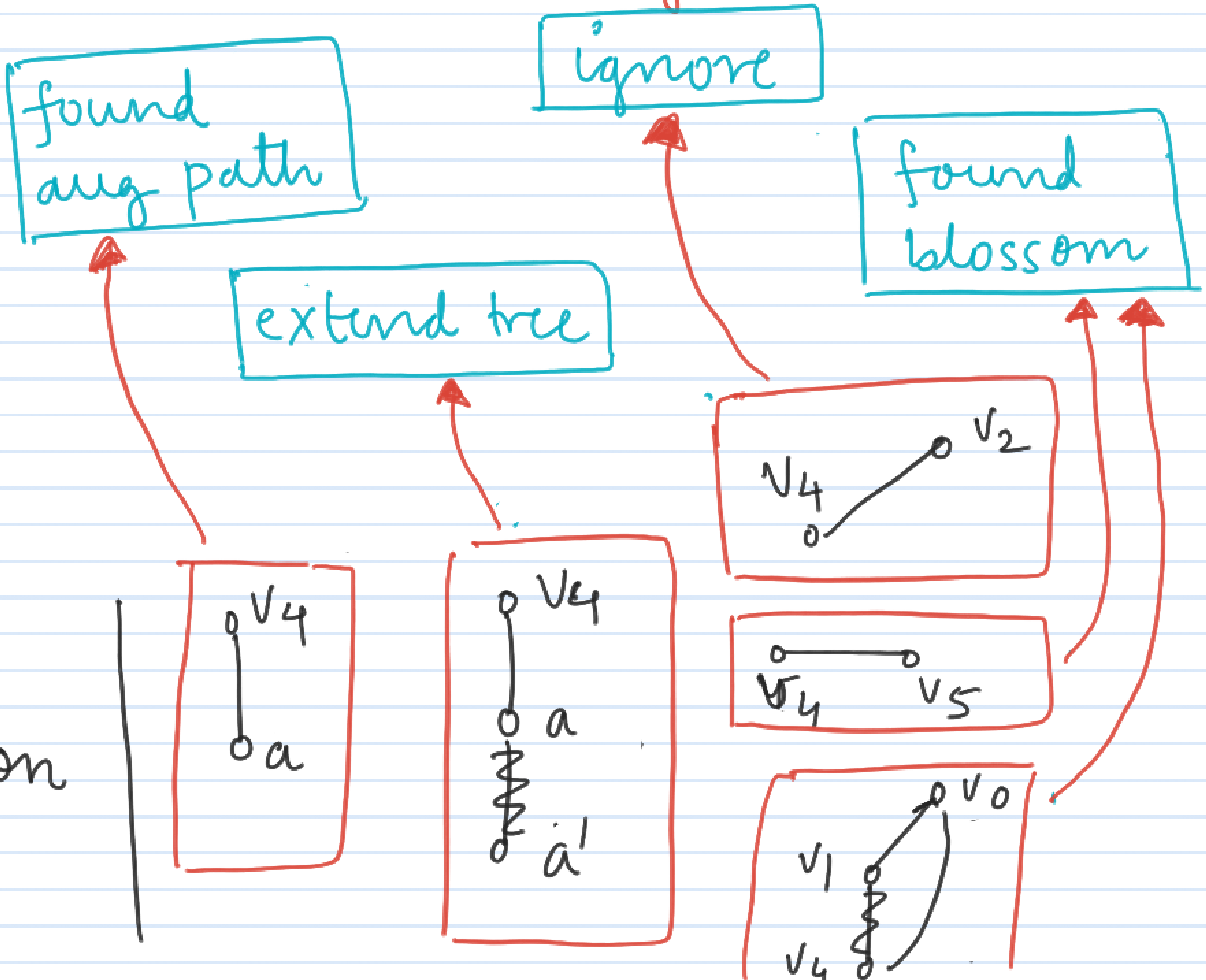
M[v] →

What kind of edges can be incident on v_4?

Q: ~~v_4~~, v_5, v_6, v_4''

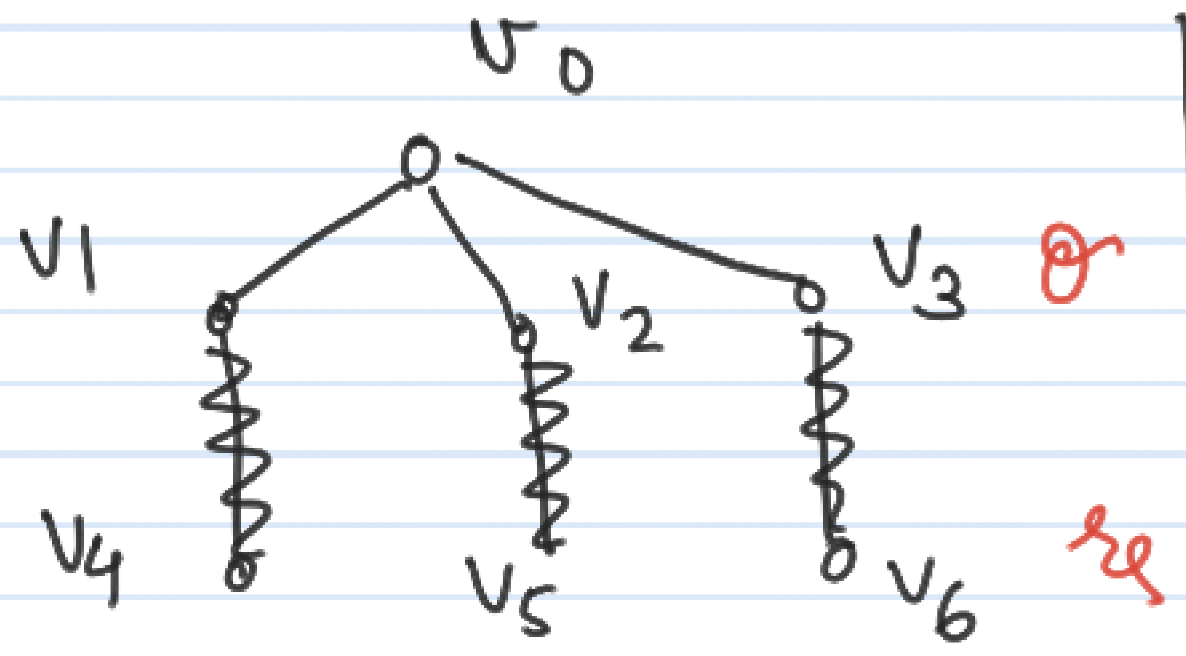
Data structures and detecting blossoms





What kind of edges can be incident on v_4?

Data structures and detecting blossoms



Global : Q of vertices

Detecting a blossom :

presence of Ev - Ev edge

What kind of edges can be incident on v_4?

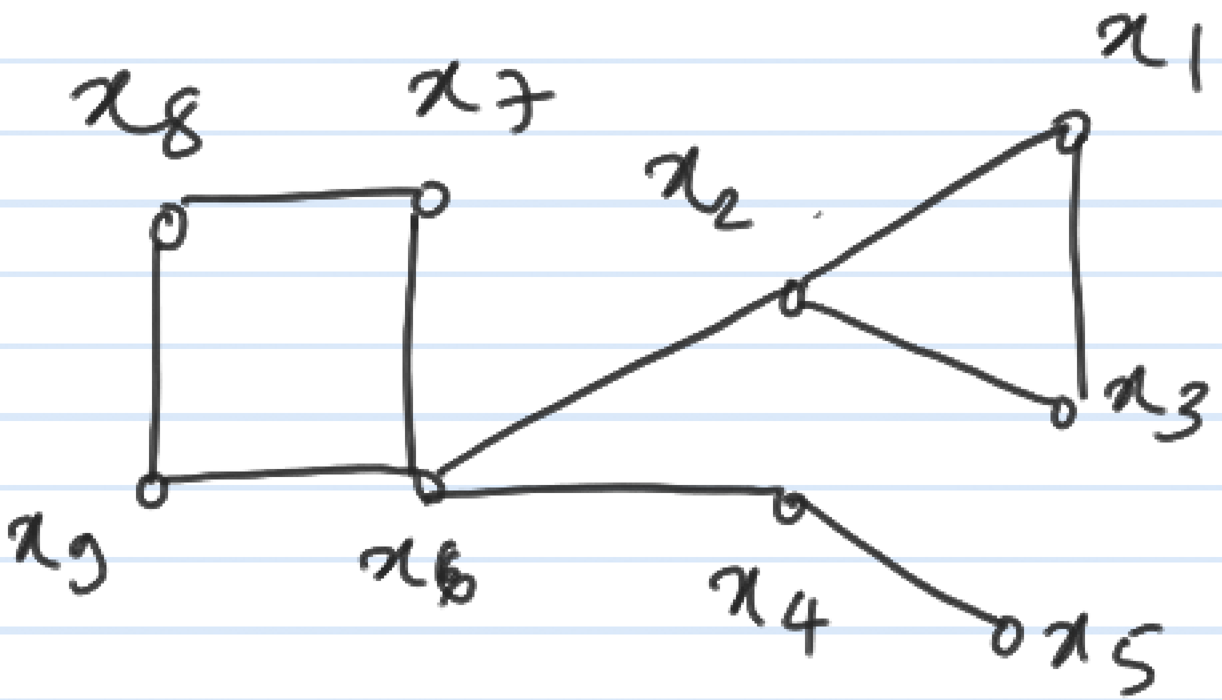
Shrinking of blossom

implicit use labels.

Reconstructing aug path

use pred[v] and labels.

Homework



G

1. compute a max matching in G

2. Label vertices as $\mathcal{O}$, $\mathcal{E}$, $\mathcal{U}$

$\mathcal{E}$, $\mathcal{O}$: defined below

$\mathcal{U} = V \setminus (\mathcal{E} \cup \mathcal{O})$.

$\mathcal{E}$: (even) : A vertex is $\mathcal{E}$ if it is reachable via an even length alt. path starting at a free vertex in M.

$\mathcal{O}$ (odd) : A vertex is $\mathcal{O}$ if it is reachable via an odd length alternating path starting at a free vertex and is not $\mathcal{E}$