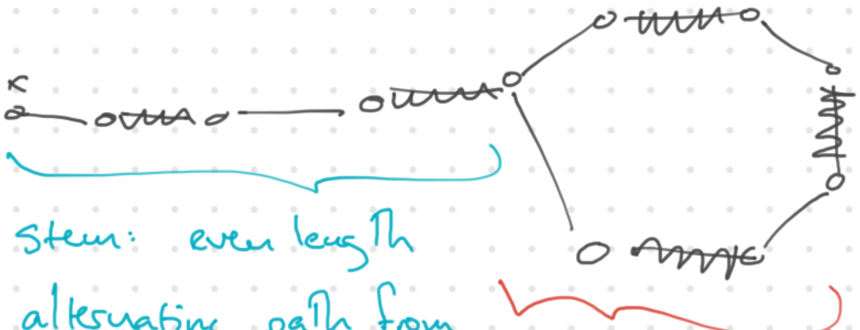
We defined an M-blossom with stem rooted at x



stem: even length alternating path from unmatched vertex x to C

blossom: ~~ev~~ odd cycle C st $\frac{|C|-1}{2}$ edges are in M

We proved: G a graph, M a matching, C an M-blossom, P a stem for C rooted at x.

Prop*: If G/C has a matching $\overline{M}$ which is strictly larger than $M - E(C)$ then G has a matching M' strictly larger than M

Algorithmic: easy to find M' given $\overline{M}$ - ~~just~~ ≤ 1 vertex x of C is incident an edge of $\overline{M}$, ~~&~~ ~~we let~~ so add $\frac{|C|-1}{2}$ edges of C to $\overline{M}$ to get a larger matching in G.

Prop If M' is a larger matching in G, then $\exists$ a matching $\overline{M}$ in G/C which is larger than $M - E(C)$

This pf was <u>not</u> algorithmic

Blossom (G a graph , M a matching) [Edmonds 61]

increase = T

While increase ≠ F Do — we grow $M \le \frac{n}{2}$ times

increase = F

For $v \in V(G)$ Do

if v is unmatched

grow BFS tree & check for augmenting paths / blossoms is $O(n+m)$ → $(P, C) = \text{Find\_aug}(G, M, x)$

returns: $C \neq 0$ an M-blossom w/ stem P or $C = \phi$, P an M-aug path or $C = \phi$, $P = \phi \Rightarrow \nexists$ M-aug. path w/ x as an end

IF $C = \text{Null} \wedge P \neq \text{Null}$

$M = M \Delta P$

increase = T

elseif $P \neq \text{Null} \wedge C \neq \text{Null}$ — depth ≤ n, $O(n(n+m))$

$\overline{M} = \text{Blossom}(G/C, M - E(C))$

if $|\overline{M}| > |M - E(C)|$

fix y to be a vertex of C not incident to an edge of $\overline{M}$ & M' a p.m. of $C - y$

$M = M' \cup \overline{M}$

increase = T

if increase = T BREAK

at end of while, return M.

Given the two propositions, the correctness of algorithm holds (easily)

FIND-AUG($G, \mathcal{M}, x$)

Q : a queue,

P : array ~~init~~ $P[y] = \text{NIL}\ \forall y \neq x$, $P[x] = x$

Q.push x

While $Q \neq \emptyset$ Do

  $v = Q$.pop()

  if $(v, P[v]) \in \mathcal{M}$ or $v = x$

    For $u \sim v$

      if $P[u] = \text{NIL}$

        Q.push(u)

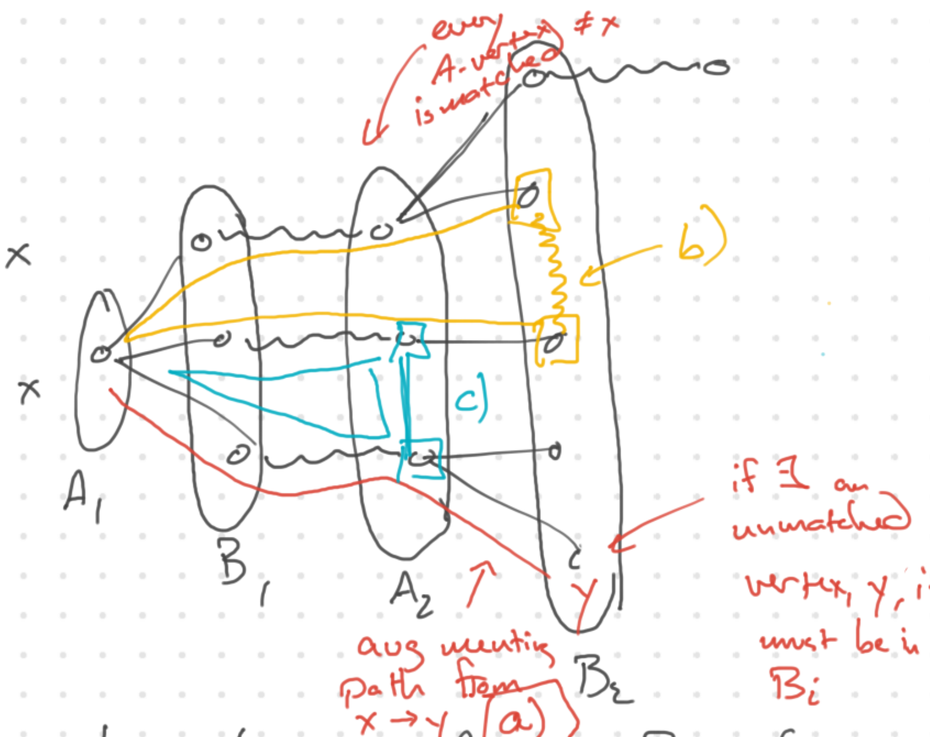
        $P[u] = v$

  elseif $(v, P[v]) \notin \mathcal{M}$

    if $\exists\ u$ ST $(u,v) \in \mathcal{M}$ + $P[u] = \text{NIL}$

      $P[u] = v$

      Q.push(u)



we have layers A_i + B_i of tree $A_i - B_i$ edges are always not in $\mathcal{M}$ $\forall i$

+ $B_i - A_{i+1}$ edges are always in $\mathcal{M}$ $\forall i$

Obs if

a) $\exists$ an unmatched leaf $\neq x \Rightarrow$
$\exists$ a $\mathcal{M}$-augmenting path

b) $\exists$ an edge $e \in \mathcal{M}$ w/ both ends in B_i for some i, $\Rightarrow$
$\exists$ an $\mathcal{M}$-blossom w/ stem + root x

c) $\exists$ an edge $e \notin \mathcal{M}$ w/ both ends in A_i for some $i \Rightarrow$
$\exists$ an $\mathcal{M}$-blossom w/ stem + root x.

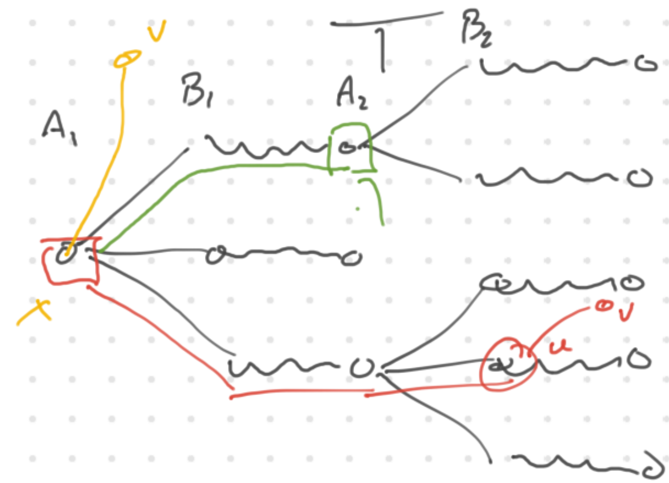
---

trace back in the tree: in first case, we get an $\mathcal{M}$-augmenting path.

In b), the paths must meet up in a vertex of A_i for some i + we get a blossom connected by an even length augmenting path to root, ie we get a stem for the blossom

Similarly, in c) we get a blossom + stem as in b)

<u>Prop</u> let T be the M-aug. BFS tree with root x. If $\exists$ an M-augmenting path w/ x as an endpoint, then one of a), b), or c) holds.



<u>pf</u> Let P be such an augmenting path & from all such augmenting paths pick it to minimize $|E(T) \cup E(P)|$

(we assume ~~so~~ $xPu \subseteq T$, $\}$ as pictured)

If $E(P) \subseteq E(T)$, then we're in case a).

So wma $E(P) \setminus E(T) \neq \emptyset$

traversing P ~~from~~ starting from x, let uv be the first edge <u>not</u> contained in T

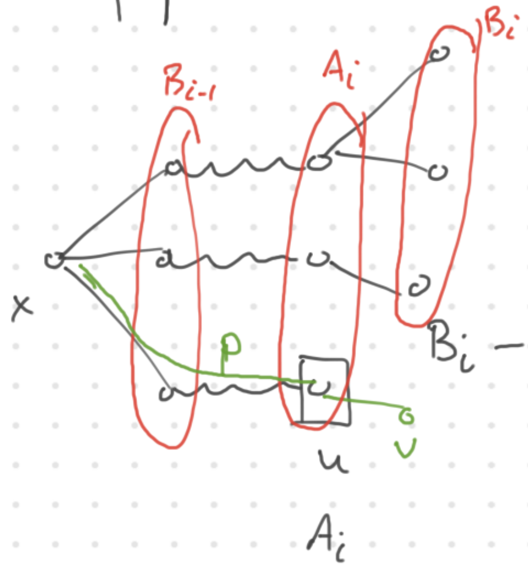
Case $u \in A_i$ for some i.

Question: can $i = 1$? No because by construction, T includes all of $N(v)$ $\Rightarrow$ $i \geq 2$

$\Rightarrow$ uv we arrived in T to

• The vertex u arriving on a matching edge from $B_{i-1} \Rightarrow uv \notin M$.

Let T' be state of tree when we pop u from the queue Q



B_i – neighbors of other A_i vertices ~~at~~ popped previously to u.

we have the edge uv in G + $uv \notin M$

$\Rightarrow v \in T'$ because our ~~...~~ edge uv would also be in T.

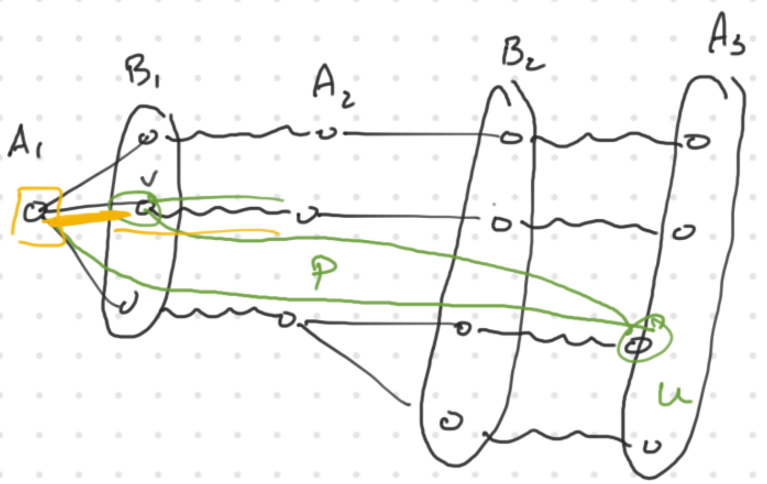
Claim: ~~$v \in B_{i-1} \cup A_i \cup B_i$~~ $v \notin A_j$ for $j < i$

if $v \in A_j$ for $j < i$ we would have popped v from Q in completing B_j + ~~...~~ since u was not in T at that point, we would have ~~...~~ included ~~...~~ uv in T $\rightarrow\leftarrow$

~~+ if $v \in B_j$ $j < i-1$~~

if $v \in A_{\geq i}$ this outcome c) + we're done.

what happens if $v \in B_j$ for $j \le i+1$

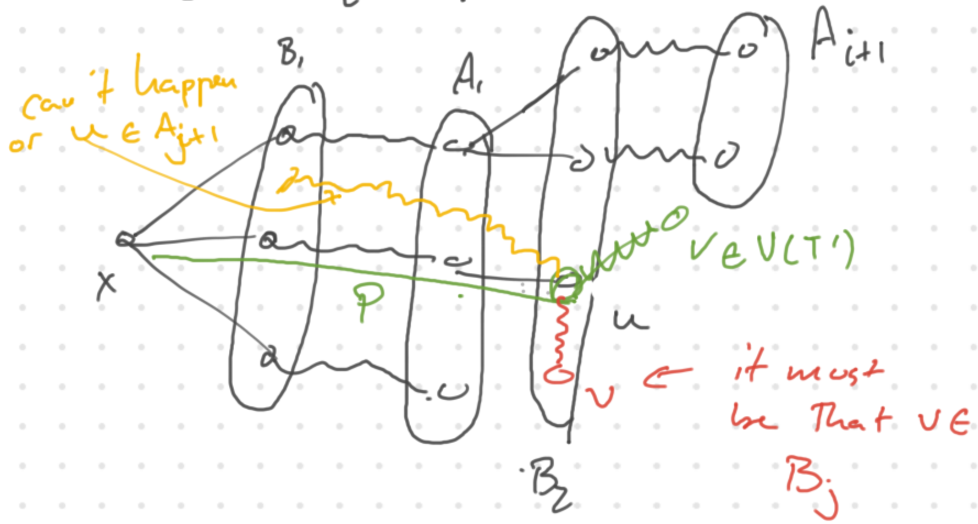


after v, P must continue on the M-edge incident to v (because we arrived at v on the $uv \notin M$).
The matching edge incident to v has it's other endpoint in A_{i+1}. (if no such matching edge exists, then v is an unmatched leaf + we are in outcome a)

So $\exists$ the matching neighbor of v in A_{i+1}. Then $xTv \cup vP$ is again an augmenting path w/ x as an endpoint, + we've avoided the edge $uv \notin E(T)$, so we've reduced $|E(T) \cup E(P')|$ for the new augmenting path

$P' = xTv \cup vP$

Case $u \in B_i$ for some i



again, let T' be the state of the tree when u is popped from Q.

P arrives at u on an edge from $A_{i-1} \to B_i$ which is not in M

Since P is M-augmenting, we have that $uv \in M$

Since we don't add uv to the tree $\Rightarrow v \in V(T')$

Note that every A_i - vertex is incident a matching edge in T by construction except for x ~~at~~ in A_1 which is not incident any matching edge.

$\Rightarrow v \in B_j \quad j \le i$

if $j < i$ - ~~At~~ Then ~~we~~ when v was popped in layer B_j, ~~we w~~ u was not yet discovered,

and we would have included
The matching edge uv between
layers B_j & A_{j+1}
so we conclude $v \in B_i$ $\Rightarrow$
we have outcome b).

This completes The pf.

Complexity $O(n(n+m))$ for each
time we grow matching for a
total of $\underline{O(n^2 m)}$