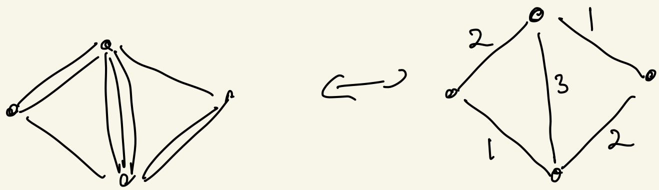
Max-back / Maximum adjacency orderings

Based on notes by Mette Eskesen (see homepage)

We consider multigraphs = parallel edges are allowed



G

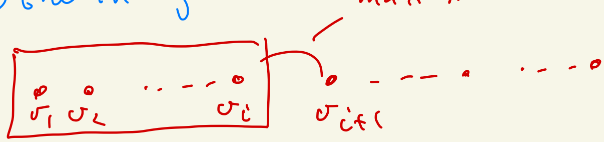
G with weights corresponding to # edges

Definition an ordering $v_1, v_2, \ldots v_n$ of the vertices of $G=(V,E)$ is called a max-back ordering of if satisfies

$$d(V_i, v_{i+1}) \geq d(V_i, v_j) \quad \text{for all } i,j \text{ with } 1 \leq i < j \leq n$$

Here $V_i = \{v_1, v_2, \ldots, v_i\}$ and $d(V_i, v_j)$ is the number of edges with one end in V_i and the other in v_j

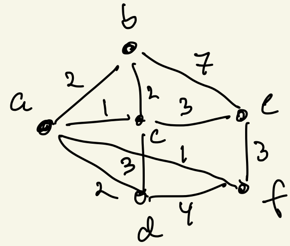
max # edges to V_i



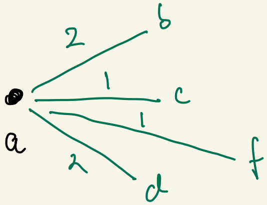
A max back ordering can be constructed in time $O(n+m)$

Easy to do in $O(n+m\log n)$ via heaps

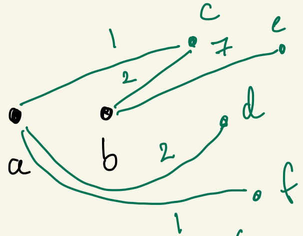
Example



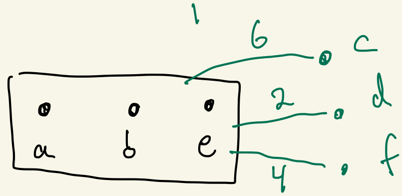
Select v_1 arbitrary. In this case a



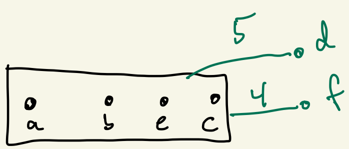
pick b as v_2 (could have taken d)



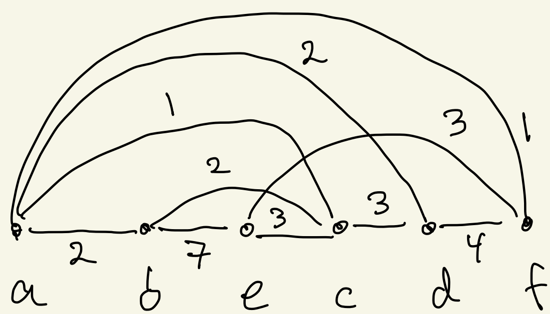
pick e as v_3



pick c as v_4



pick d as v_5 (and f as v_6)

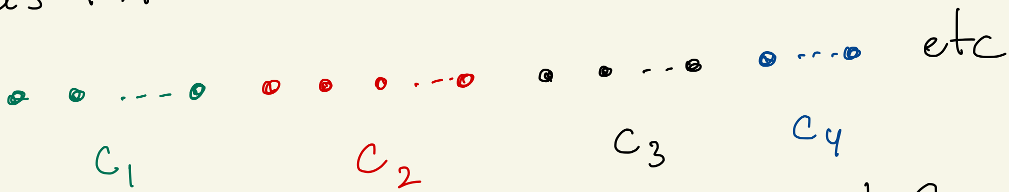


Final ordering

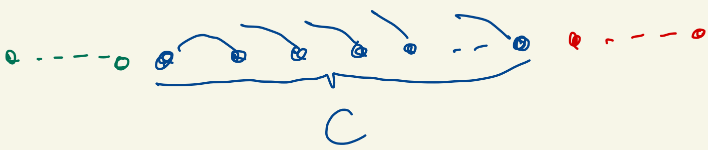
Continous orderings

Definition An ordering $v_1, v_2 \cdots v_n$ is continous if it satisfies

(i) The connected components of G occours as intervals in the ordering



(ii) Every vertex of a connected component C has an edge to a lowered numbered vertex in C except for the vertex with the lowest number C

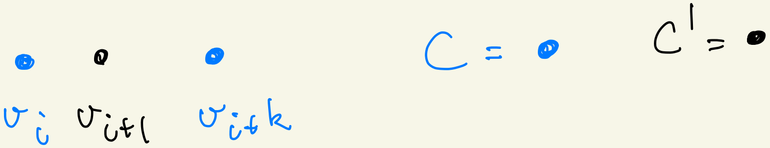


Easy: If $v_1, v_2 \cdots v_n$ is continous and $d(v_n) > 0$ then v_n and v_{n-1} are in the same connected component of G.

Lemma 2 Every max-back ordering is continuous

proof: Suppose (i) does not hold for the m.b.o. $v_1, v_2, \ldots, v_n$ of G.

Then G has at least 2 connected components and there is a smallest $i \in [n]$ and a component C such that $v_i, v_{i+k} \in C$ for some $k \geq 2$, but $v_{i+1} \notin C$



By the choice of i, C' has no vertex in $v_1, v_2, \ldots, v_i$ so $d(V_i, v_{i+1}) = 0$

However v_i and v_{i+k} are connected by a path in C so $d(V_i, v_j) > 0$ for some $j \in \{i+2, \ldots, n\}$

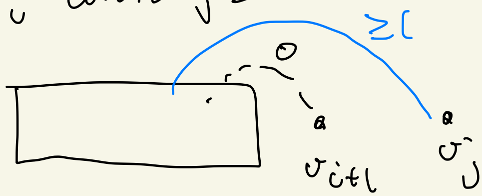
Now $d(V_i, v_{i+1}) < d(V_i, v_j)$ $\rightarrow\leftarrow$ as $v_1, v_2, \ldots, v_n$ is a m.b.o.

This shows that (i) holds for every max-back ordering.

(ii): Suppose $C = \{v_k, v_{k+1}, \dots v_\ell\}$ is a component and that $v_{i+1} \in C$ has no edge to a lower numbered vertex in C.

$i+1 \neq \ell$ as C is connected (all edges incident to v_ℓ go to lower vertices)

Now $d(V_i, v_{i+1}) = 0$ by every path from v_ℓ to v_k in C contains an edge from some v_j to V_i when $j \geq i+2$.



Now $d(V_i, v_{i+1}) < d(V_i, v_j) \longrightarrow \Leftarrow$.

This concludes the proof of (ii) and the lemma. $\square$

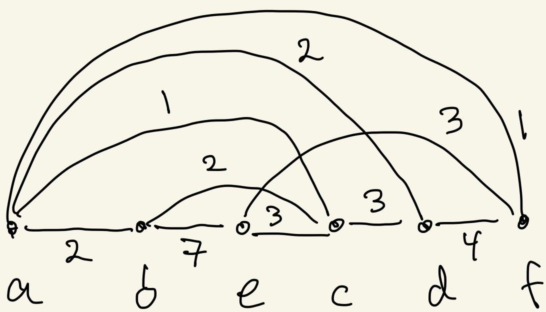
<u>Definition</u> Let $\Omega = v_1, v_2, \ldots, v_n$ be a m.b.o. of $G=(V,E)$

The max-back forest $F = F(\Omega)$ corresponding to Ω is given by $F = (V, E')$ when

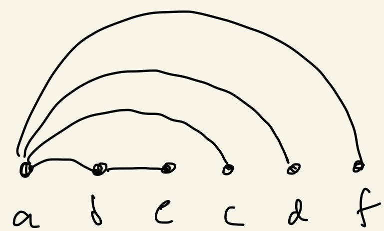
$E' = \{ v_i v_j \mid j > i \wedge v_i v_j \in E \wedge d(V_{i-1}, v_j) = 0 \}$

So for each vertex $v_2, v_3 \ldots v_n$ we take the furthest edge back (if any)

G



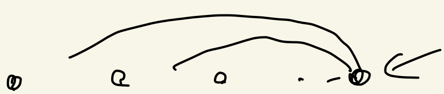
Ω



$F(\Omega)$

$|E'| \le n-1$ as ≤ 1 edge for each of $v_2 \ldots v_n$

E' is acyclic: if cycle D then



max # vertex in D

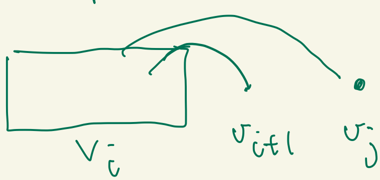
Note: $|E'| = |V| - \#$ components of G
(only first vertex of each component has no edge in E')

$\Rightarrow$ E' contains a spanning tree of each connected component of G.

Lemma 3 Let $v_1, v_2 \cdots v_n$ be a m.b.o. and let F be the corresponding m.b.f.
Then $v_1, v_2, \cdots v_n$ is also a m.b.o. of $G' = G - E'$

proof: suppose not. Then $\exists\, i, j$ $i+1 < j$ s.t

In G':



$d_{G'}(V_i, v_{i+1}) < d_{G'}(V_i, v_j)$

This implies that $d_{G'}(V_i, v_j) \geq 1$
so we also have $d_G(V_i, v_j) \geq 1$
As $v_1, v_2 \cdots v_n$ is an m.b.o. of G we have $d_G(V_i, v_{i+1}) \geq d(V_i, v_j) \geq 1$ and then

$$d_{G'}(V_i, v_{i+1}) = d_G(V_i, v_{i+1}) - 1$$
$$\geq d_G(V_i, v_j) - 1 = d_{G'}(V_i, v_j)$$

Recall that we denote by $\lambda(x,y)$ the local edge-connectivity between x and y

By Menger's theorem we have

$$\lambda(x,y) = \min\{d(X) \mid x \in X, y \in V-X\}$$

<u>Theorem 4</u> For every m.b.o. $v_1, v_2, \ldots, v_n$ of G we have $\lambda(v_{n-1}, v_n) = d(v_n)$

<u>Proof:</u> Clearly $\lambda(v_{n-1}, v_n) \leq d(v_n)$

If $d(v_n)=0$ then is nothing to prove so assume that $d(v_n)=k>0$

let F_1 be the max-back forest of G wrt $v_1, v_2 \ldots v_n$

By lemma 3 $v_1, v_2 \ldots v_n$ is also a m.b.o of $G-E_1$ when $E_1 = E(F_1)$

Now let $G_1 = G$, $G_2 = G - E_1$ and $G_i = G - \bigcup_{j<i} E_j$

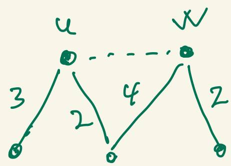
when F_j is a m.b.f of $G_j = G - \bigcup_{q<j} E_j$ and $E_j = E(F_j)$

Then $d_{F_j}(v_n) = 1$ for $j = 1, 2 \ldots, k$ so v_{n-1} and v_n are in the same component of $F_1, F_2 \ldots F_k$ as $v_1, v_2 \ldots v_n$ is continuous.

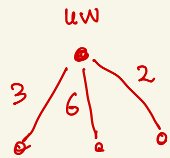
This implies that v_{n-1} and v_n are in the same tree of F_i for $i = 1, 2 \ldots k$ so $\lambda(v_{n-1}, v_n) \geq k$ □.

Determining the edge-connectivity via m.b.o.'s

Let u and w be distinct vertices of $G=(V,E)$. Then we denote by G^{uw} the graph obtained by identifying u and w



in G



keep all edges except loops at uw

The edge-connectivity of G, denoted $\lambda(G)$ is

$$\lambda(G) = \min\{\lambda(x,y) \mid x,y\in V,\ x\neq y\} \qquad (*)$$

Theorem 5 Let $G=(V,E)$ be a multigraph with $n\geq 3$ vertices, then for every pair of distinct vertices $u,v\in V$

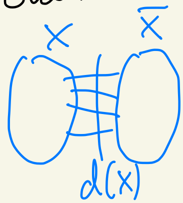
$$\lambda(G) = \min\{\lambda(u,v),\ \lambda(G^{uv})\}$$

Proof:

$\lambda(G) \leq \lambda(u,v)$ clear from $(*)$

$\lambda(G) \leq \lambda(G^{uv})$ clear as G^{uv} has fewer cuts

Assume $\lambda(G) = d(X)$



if $|X\cap\{u,v\}|=1$ then $\lambda(u,v)\leq d(X) = \lambda(G)$

if not then $\lambda(G^{uv})\leq d(X) = \lambda(G)$ $\square$.

Algorithm for finding $\lambda(G)$

If $n = |V| = 2$ return $d(v)$ v arbitrary

$K \leftarrow \infty$, $G_0 \leftarrow G$, $i \leftarrow 0$

Repeat

Let $v_1, v_2 \cdots v_{n-i}$ be a m.b.o of G_i

$K \leftarrow \min \{K, d_{G_i}(v_{n-i})\}$

$G_{i+1} \leftarrow G_i^{v_{n-i-1} v_{n-i}}$

$i \leftarrow i+1$

Until $i = n-2$

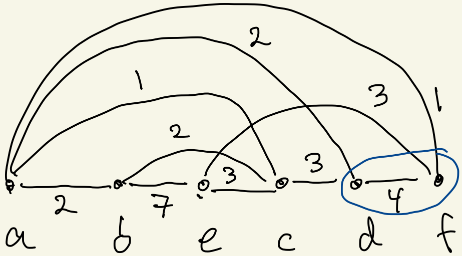
Return K

Theorem 5 $\rightarrow$ output is $K = \lambda(G)$

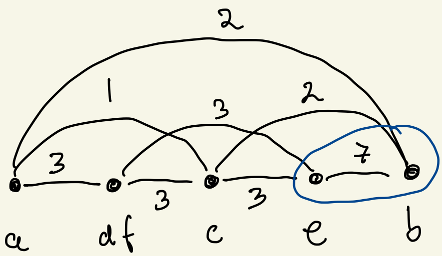
Running time $O(n) \cdot$ time for m.b.o

so $O(n(n+m))$

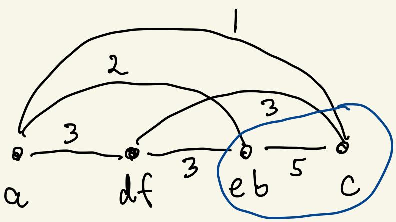
Finding the edge-connectivity of our example



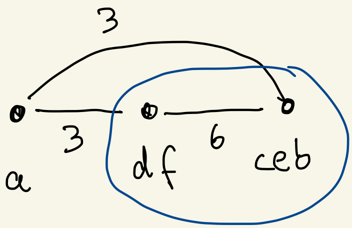
a, b, e, c, d, f is a MAO and $d(f) = 8$ so $K \leftarrow 8$ and d, f identified



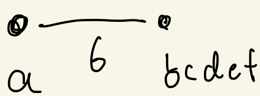
$d(b) = 11 > K$ so K unchanged
e and b identified



$d(c) = 9 > K$ so K unchanged
eb and c identified



$d(ceb) = 9 > K$ so K unchanged
df and ceb identified



$d(bcdef) = 6 < K$ so $K \leftarrow 6$

Return $\lambda(G) = 6$

Sparse certificates for edge-connectivity

Lemma 7

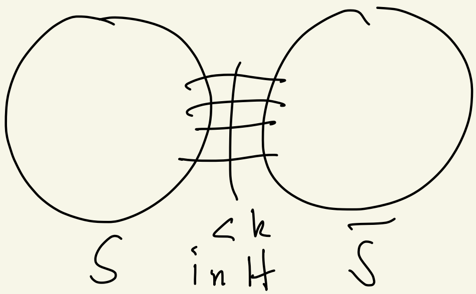
Suppose $\lambda(G) = k$ and $\Omega = \sigma_1, \sigma_2, \ldots \sigma_n$ is a m.b.o of G.

Let F_i be the max-back forest of $G_i = G - \bigcup_{j<i} E(F_j)$ and all F_i's are wrt Ω.

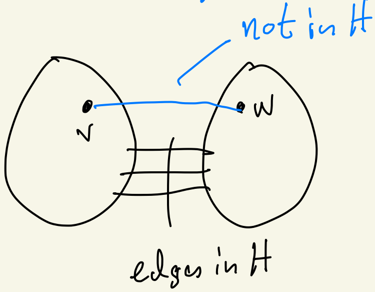
Then $H = (V, \bigcup_{i=1}^{k} E(F_i))$ is k-edge-connected.

Proof:

Suppose $\lambda(H) < k$ and let $S \subseteq V$ be such that $d_H(S) < k$



As $\lambda(G)=k$ there is at least one edge vw with $v\in S$, $w\in\bar{S}$ and $vw\notin E(H)$



$vw \in E(G)-E(H)$

$\Downarrow$ as each F_i is maximal (spT in each component of G_i)

v and w are in the same connected component of F_i for $i=1,2,\dots,k$

$\Downarrow$

$\lambda_H(v,w)\geq k \quad \rightarrow\leftarrow$

Algorithm to construct H:

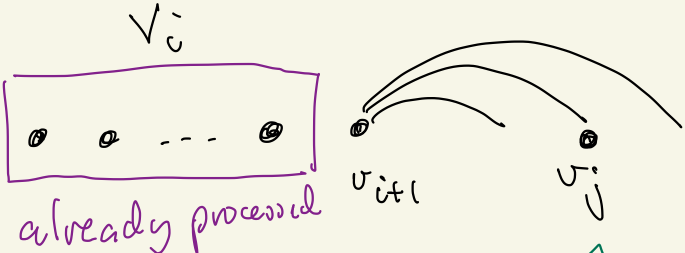
Find a m.b.o. $\Omega = v_1, v_2, \dots, v_n$

For each $i=2,3,\dots,n$ take the $\min\{k, d(V_{i-1}, v_i)\}$ furthest edges back from v_i

How to do this efficiently:

First find m.a.o $v_1, v_2, \ldots v_n$

scan forward $\rightarrow$

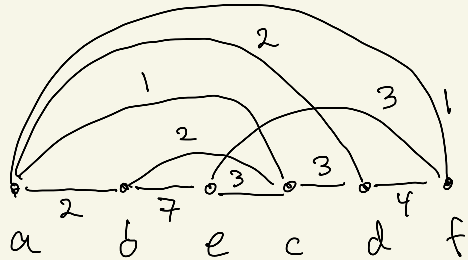


if $d(V_i, v_j) = r < k$
add $\min\{k-r, d(v_{i+1}, v_j)\}$
edges from v_j to v_{i+1}

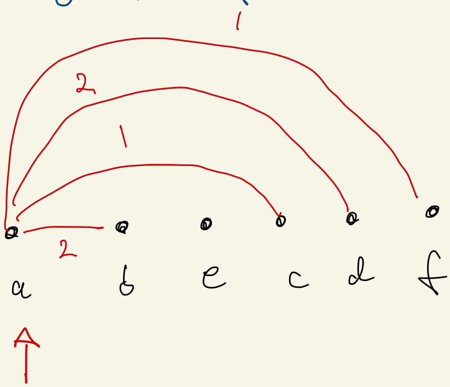
Corollary

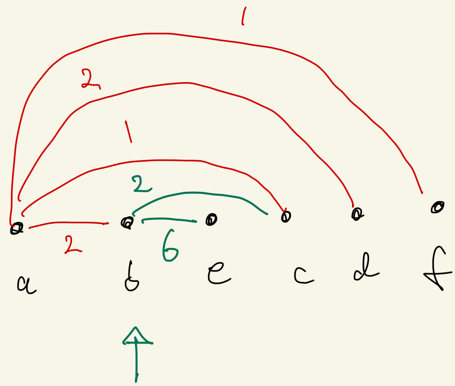
Every graph G with $\lambda(G) = k$ has a spanning subgraph H s.t. $\lambda(H) = k$ and $E(H) \leq k(n-1)$

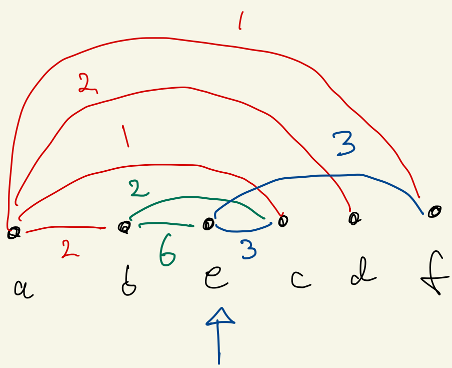
G

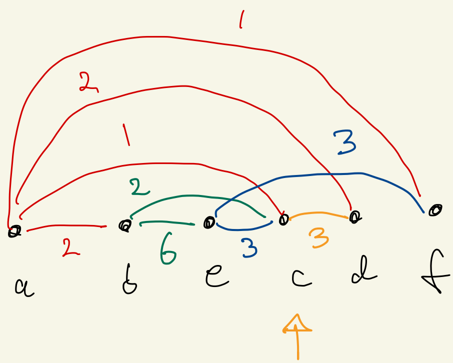


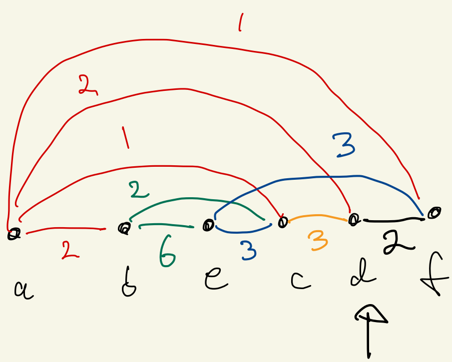
We found $\lambda(G) = 6$ so we get 6 m.b.f's



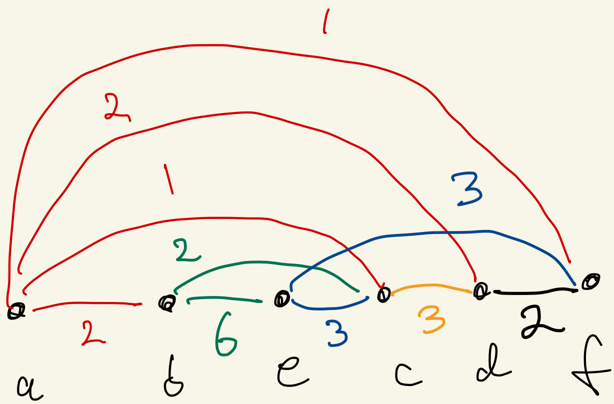








Resulting spanning 6-edge-connected subgraph H:



$$|E(H)| = 6+8+6+3+2 = 25 < 30 = 6\cdot(6-1)$$