General matching

Recall: Given an undirected graph, a matching is a set of edges that do not share vertices.

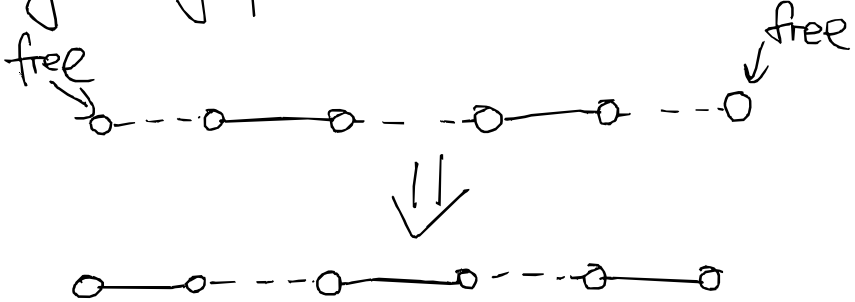
Problem: Compute maximum matching of the input graph.

Today: general graph $G = (V, E)$.

Main idea: improve the matching size via augmenting paths.

Last time: "max matching iff no aug path" holds for general graphs

Augmenting path

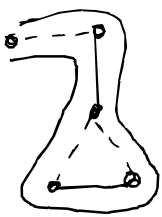


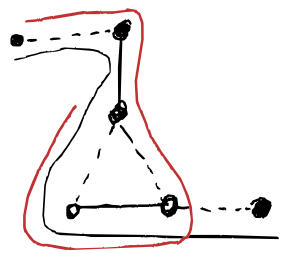
Main difficulty for general graphs:

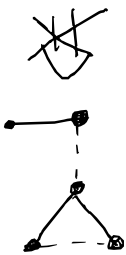
find an augmenting path efficiently

Why doesn't DFS work?

- for bipartite $G=(X \cup Y, E)$,
  - for $x \in X$, DFS(x) returns an alternating path from x starting with an unmatched edge ending at a free vertex, or returns no such path.
  - every time x is reached during DFS, next edge must be unmatched (every cycle has even length).
    No reason to visit a visited vertex again.
- for general graphs, there may be odd cycles.



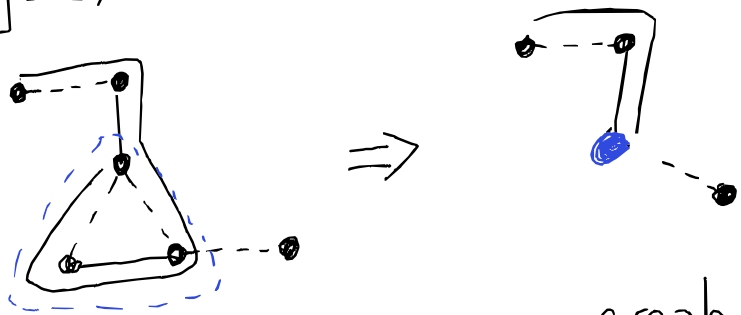




  - DFS from x may want to start with unmatched or matched edge
  - Augmenting path must not contain the same vertex more than once.

Blossom algorithm

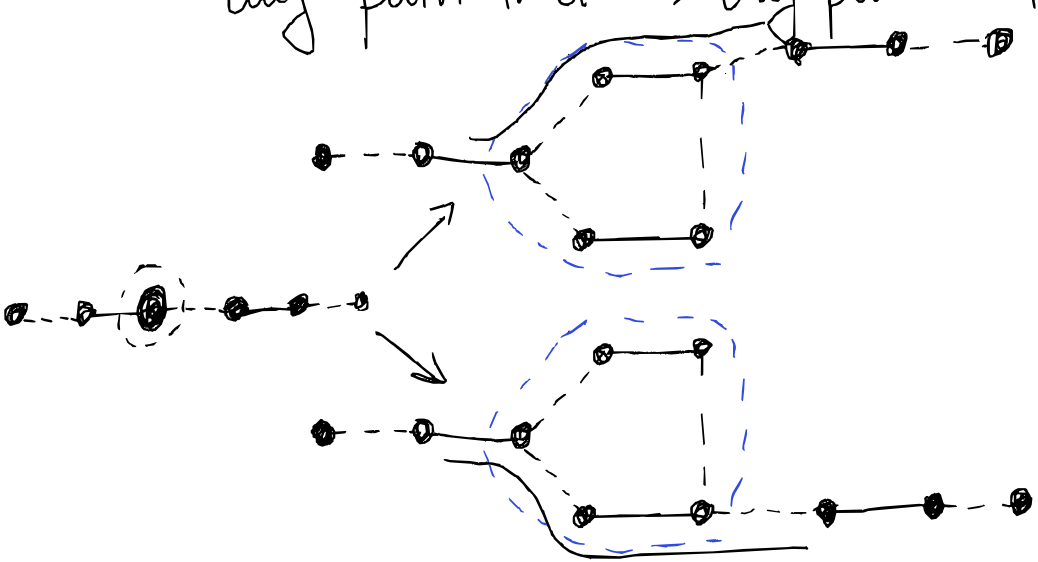
Each time traverse an odd alternating cycle, contract it into a supernode.



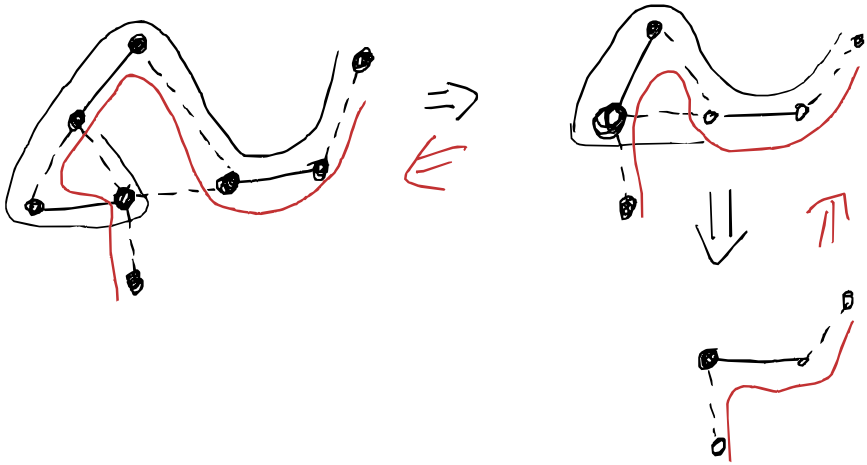
old graph G new graph G'

Search for augmenting paths in G'

- aug path in $G' \Rightarrow$ aug path in G

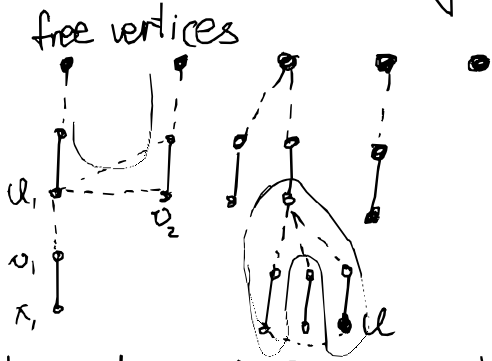


Blossom may be nested:



The algorithm:

- maintain search trees from all free vertices with alternating paths



- start with all free vertices as roots
- expand search trees in any order:

for each unprocessed vertex u, and all its incident (unmatched) edges (u,v):

case I: v is not in the trees
(v,x) is a matching edge,
add $(u,v), (v,x)$ to the trees

case II: v is in a different tree than u
v has an even distance to its root
found an augmenting path

case III: v is in the same tree as u
v has an even distance to its root
found a blossom, contract and continue

case IV: v has an odd distance to its root
(do nothing)

Running time to find an aug path:

case I, IV: $O(1)$ per edge

case II: $O(n)$ time to unfold all blossoms and find the path. occur only once.

case III: $O(n)$ time to find the blossom, and to "relabel" the vertices in the blossom to the new supernode, and contract. Occurs at most n times.

$O(n^2+m) = O(n^2)$ time.

Total time: $O(n^3)$.