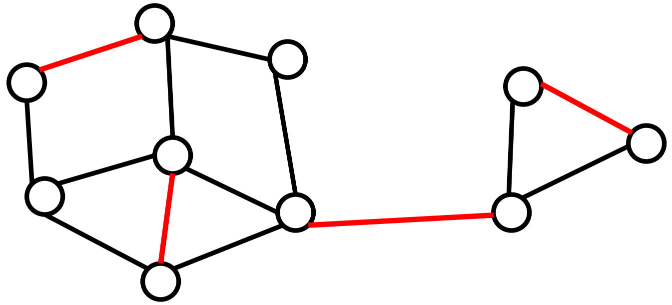
Matchings: Max Cardinality and Min Cost

CSC 473 Advanced Algorithms

Computer Science
UNIVERSITY OF TORONTO

Matchings in graphs

- A matching in a graph $G = (V, E)$ is a subset $M \subseteq E$ of edges so that no two edges in M share an endpoint.



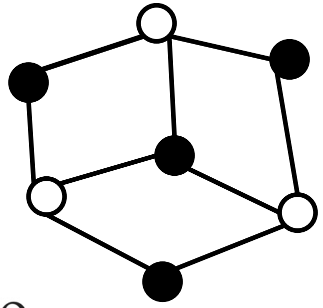
- Maximum cardinality matching: given input graph G, find a matching M of maximum size
  - Perfect matching: size $\frac{|V|}{2}$, i.e., all edges are matched
  - Solvable in polynomial time

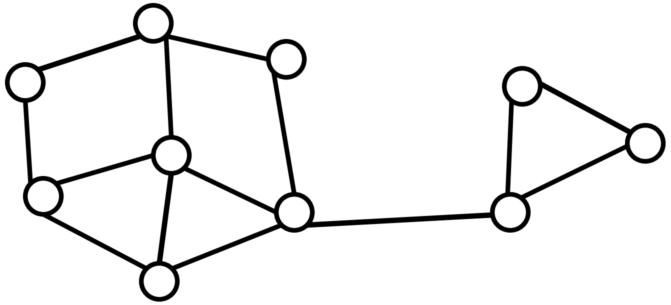
Bipartite Graphs

Algorithm for general graphs is a deep result of Jack Edmonds.

- We will focus on max cardinality matching in *bipartite graphs*.
- Bipartite graph: $G = (V, E)$ so that we can partition V into disjoint sets A and B, and all edges in E have one endpoint in A and on in B
  - We can check if G is bipartite in time $O(n + m)$
  - If it is, we can also find A and B in this time
- *Fact*: a graph is bipartite if and only if it does *not* have an odd cycle
  - *only if* : any path alternates A and B and can only come back to the starting node after an even numbers of hops.

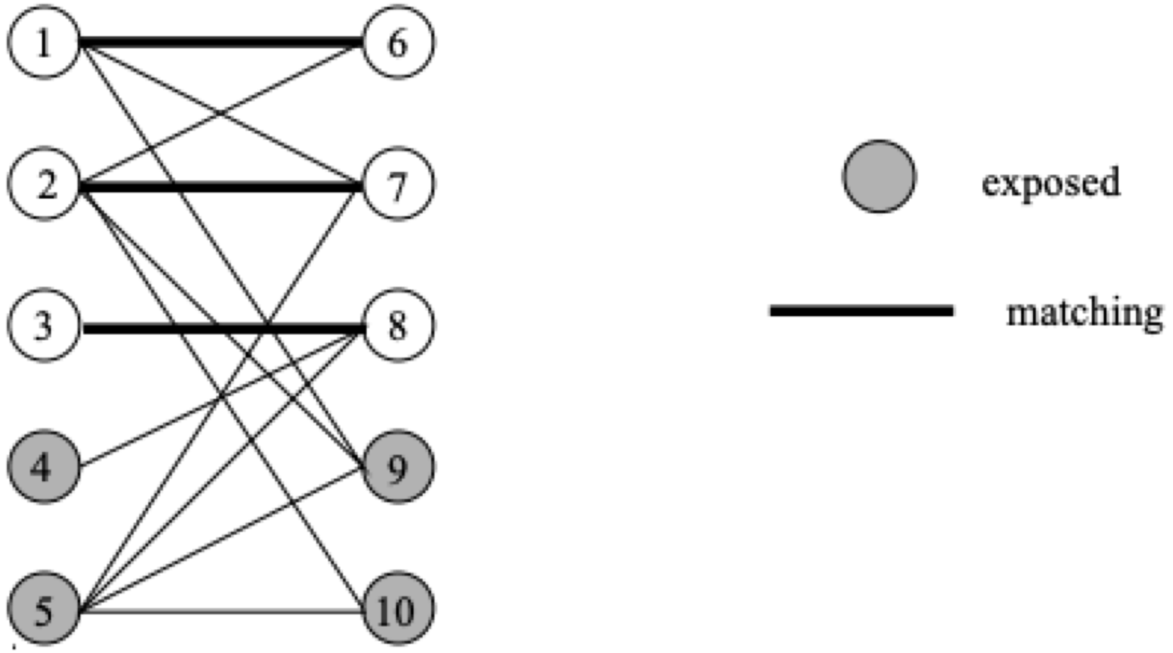
Bipartite





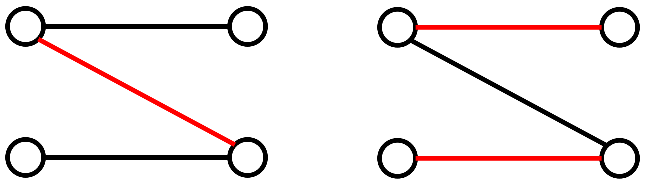
Not bipartite

Bipartite Matching

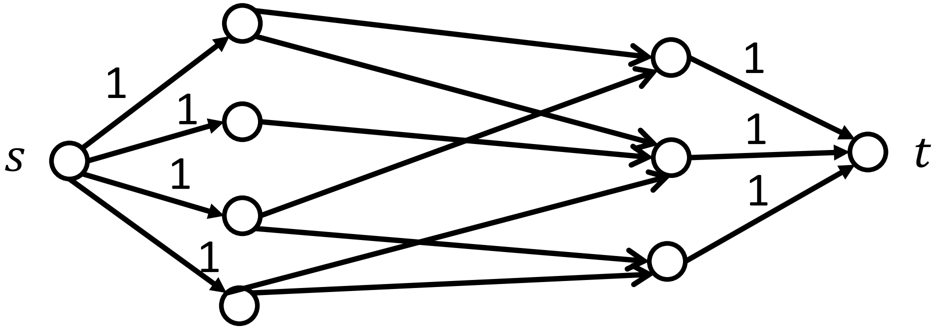


Finding the Maximum Matching

- Greedy (keep adding edges while you can) does not work

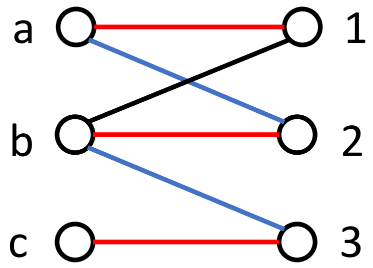


- Do you know a polynomial time algorithm to find the max matching?
  - Compute a max flow
  - We will see a more combinatorial algorithm

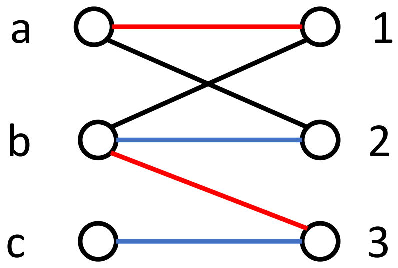


Augmenting Paths

- For a bipartite graph G and a matching M, a path P is <u>alternating</u> if edges in P alternate between being in M and being outside of M.
- An alternating path is <u>augmenting</u> if it starts and ends in unmatched vertices.



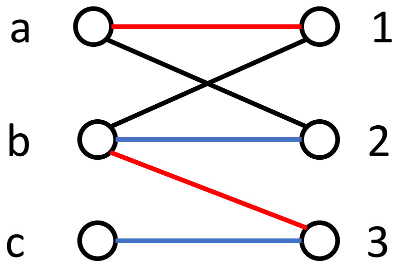
1 -> a -> 2 -> b -> 3 -> c
is alternating



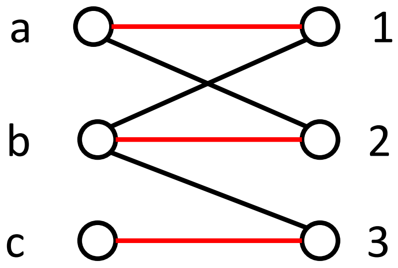
2 -> b -> 3 -> c
is augmenting

Augmenting Paths

- An alternating path is augmenting if it starts and ends in unmatched vertices.
- Let P = augmenting path. Set $M' = M \triangle P = (M \cup P) \setminus (M \cap P)$.
- M' is a matching and $|M'| = |M| + 1$



2 -> b -> 3 -> c
is augmenting



Flip which edges are in M along P.

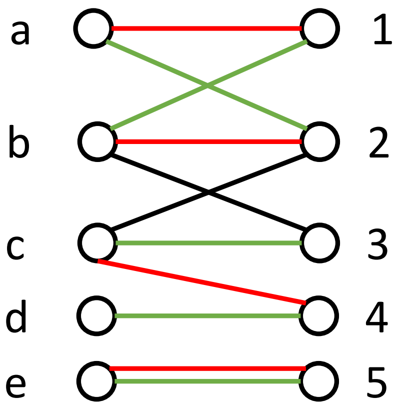
First and last vertex in P unmatched, and others just switch which ones they are matched to.

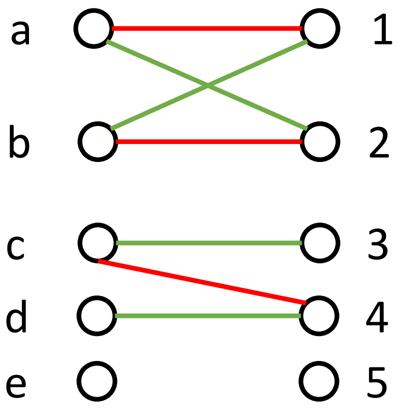
One more non-matching edge in P than matching edges: size of matching increases by 1.

Characterizing Max Matchings

Theorem. A matching M is of maximum cardinality if and only if there is no augmenting path for it.

- only if: Augmenting path means there is a larger matching
- if: Take a matching M', $|M'| > |M|$, and graph H with edges $M \triangle M'$
  - $M \triangle M' = (M \cup \mathrm{M}') \setminus (M \cap M')$





- H has max degree ≤ 2
- The connected components of H are alternating paths and even cycles
- H has more edges from M', so has a path component starting and ending with edges from M': augmenting for M.

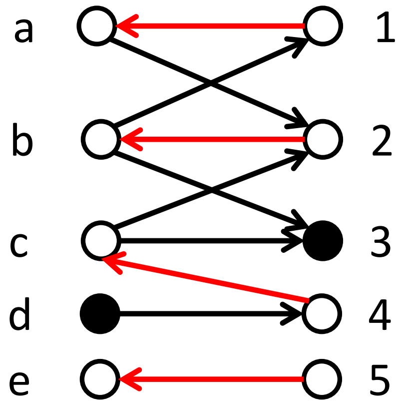
High-Level Algorithm

- $M = \emptyset$
- While $\exists$ an augmenting path P for M
  - Set $M = M \triangle P$

- Correct by Theorem.
- At most $\frac{n}{2}$ iterations, since each iteration adds an edge to M
- How do we search for an augmenting path?
  - Will show a $O(n + m)$ time algorithm, for $O(n^2 + nm)$ total time

Finding Augmenting Paths

- Idea: construct a directed graph $G_M = (V, E_M)$
- alternating paths in G <--> directed paths in G_M



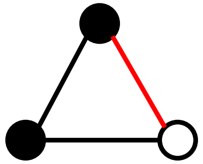
- Direct edges in M from right to left
- Direct edges not in M from left to right
- Any path in G_M must alternate edges in and outside M
- Use BFS to search for a path in G_M from an exposed vertex on the left to an exposed vertex on the right.

König’s Theorem

- Vertex Cover: a set C of edges of $G = (\mathrm{V}, \mathrm{E})$, so that for any edge $(a, b) \in E$, $a \in C$ or $b \in C$ (or both)

Theorem. In any bipartite graph, the size of the minimum cardinality vertex cover equals the size of the maximum cardinality matching.

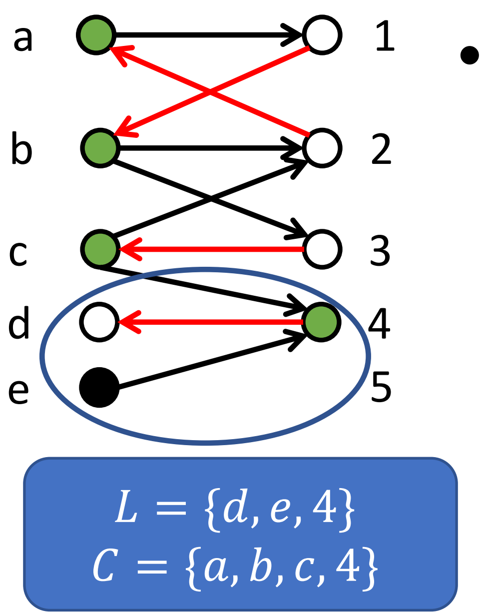
- Easy: for any v.c. C and matching M, $|M| \leq |C|$
  - Edges in M are disjoint, so no vertex in C can cover more than one of them
  - True even for non-bipartite graphs.
- Harder: for the max matching M, there exists a v.c. C s.t. $|C| = |M|$
  - This part fails for some non-bipartite graphs.



Computer Science
UNIVERSITY OF TORONTO

Proof of Harder Direction

- M = max matching (no augmenting path)
- U = exposed vertices; L = vertices reachable in G_M from $U \cap A$
- $C = (A \setminus L) \cup (B \cap L)$ is a vertex cover of size $|C| = |M|$

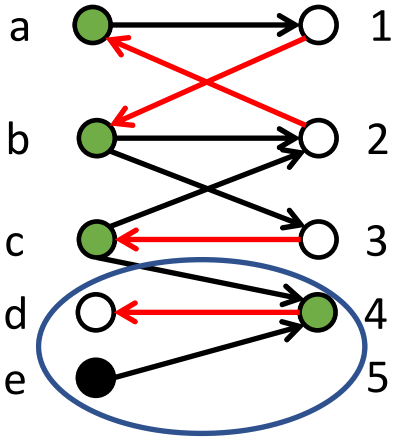


- No edges (a, b) with $a \in A \cap L$ and $b \in B \setminus L$
  - $(a, b) \notin M$: if a is reachable from U, then so is b
  - $(a, b) \in M$: $a \notin U$, and only incoming edge is $a \leftarrow b$, so a is reachable from U only if b is

Computer Science
UNIVERSITY OF TORONTO

Proof, continued

- M = max matching (no augmenting path)
- U = exposed vertices; L = vertices reachable in G_M from $U \cap A$
- $C = (A \setminus L) \cup (B \cap L)$ is a vertex cover of size $|C| = |M|$



$L = \{d, e, 4\}$
$C = \{a, b, c, 4\}$

- Every vertex in C touches exactly one edge in M
  - $A \setminus L \subseteq A \setminus U$ so all vertices in $A \setminus L$ are matched
  - All vertices in $B \cap L$ are matched, otherwise there is an augmenting path.
  - If $(a, b) \in M$, and $b \in B \cap L$, then $a \notin A \setminus L$ (if b is reachable from U, so is a).