Algorithms on Graphs

Module 3

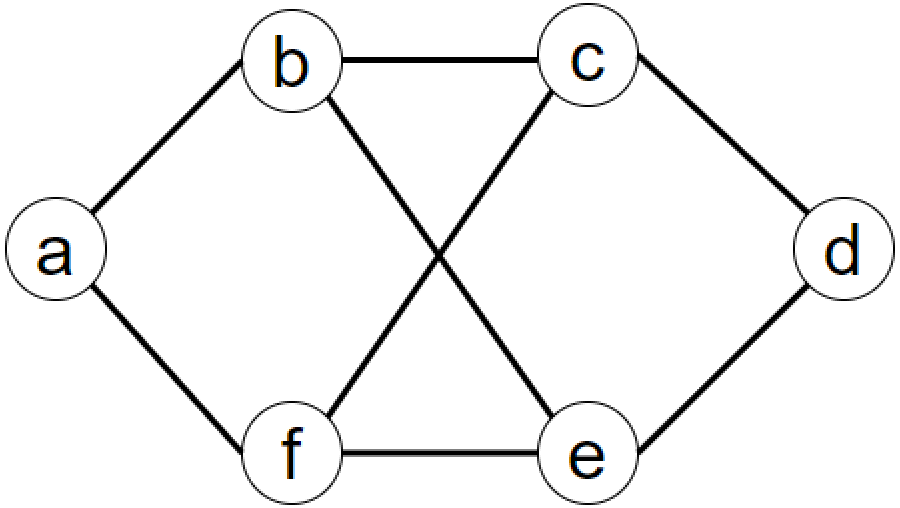
Lecture 11
Matchings

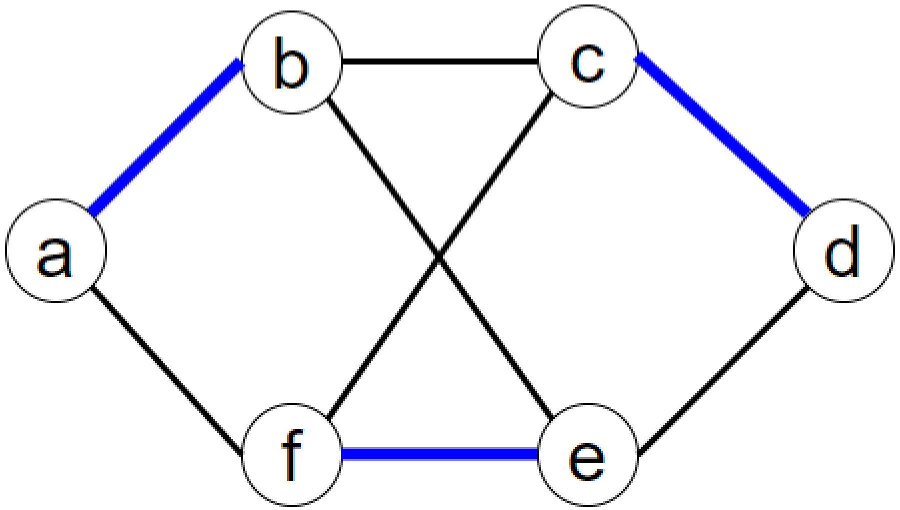
Adigeev Mikhail Georgievich

mgadigeev@sfedu.ru

Matchings

Definition. ***Matching (match)*** on a graph $G(V, E)$ is a subset of edges $M \subseteq E$ such that no two share a vertex.

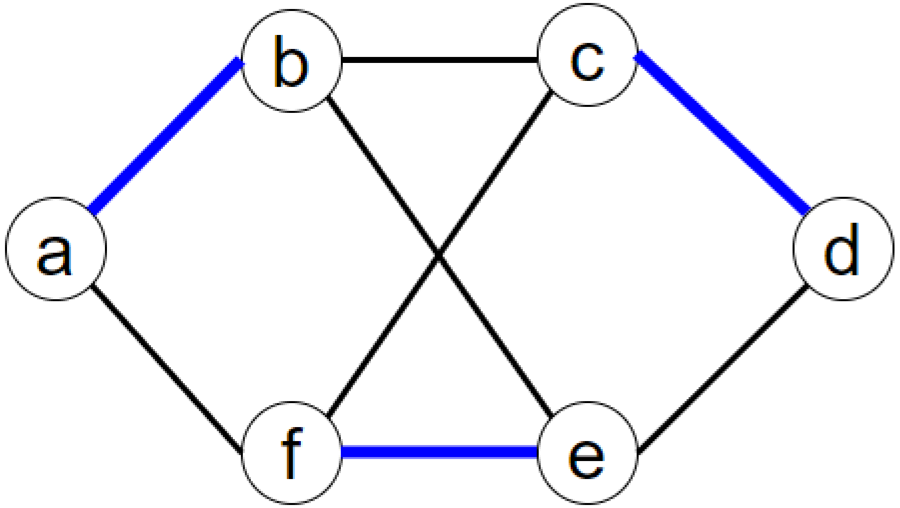




Matchings

- **Definition**. ***Perfect matching*** is a matching that covers all vertices of the graph. A vertex is called covered by an edge if they are incident.
- **Definition**. A matching is called ***maximal*** if the graph has no matching with greater cardinality (number of edges).

It is evident that any perfect matching is maximal, but the converse is not true in general.

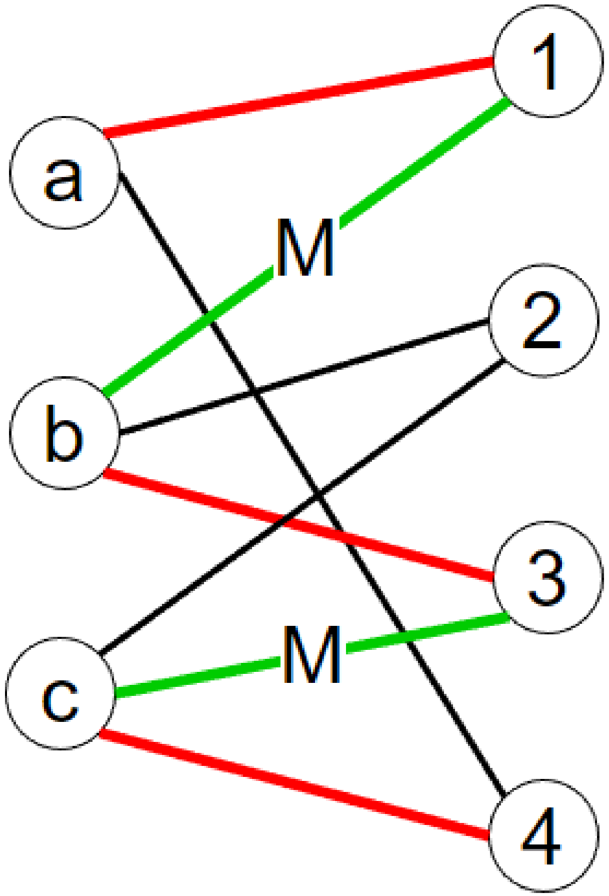


Matchings

Let M be a matching on the graph $G(V,E)$.

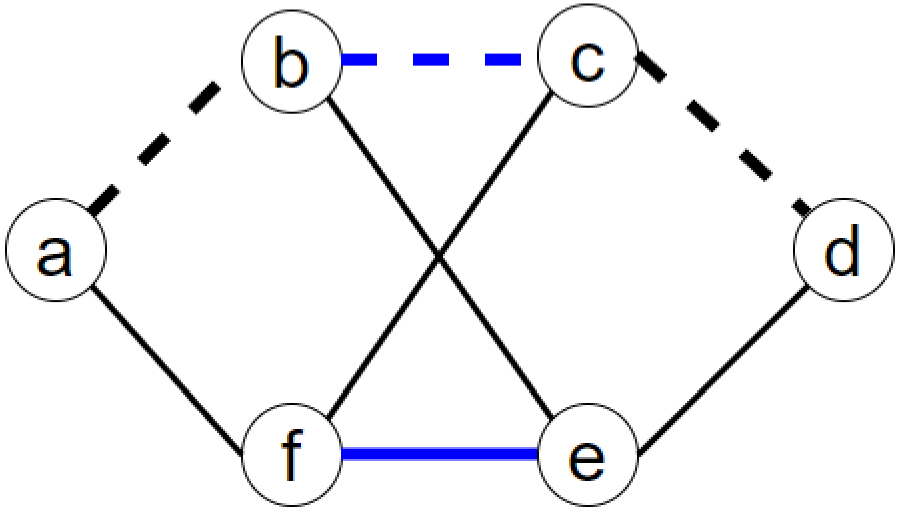
- **Definition**. ***An alternating path*** for M is a path that alternates between edges in M and edges not in M.
- **Definition**. Vertex v is called ***matched (saturated)***, if it is incident to some edge in M, and is called ***unmatched (exposed)*** otherwise.
- **Definition**. An alternating path is called an ***augmenting*** path iff both of it's endpoints are unmatched.

NB: all these notions are with respect to the given M.



Matchings

Statement. An augmenting path contains odd number of edges, and the number of matched edges is 1 greater than the number of unmatched edges.



Matchings

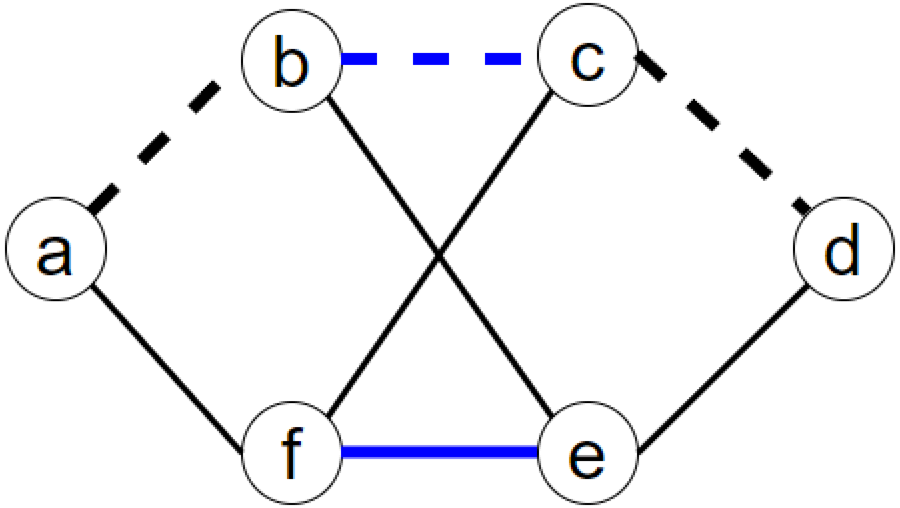
Let M be a matching on graph $G(V, E)$.

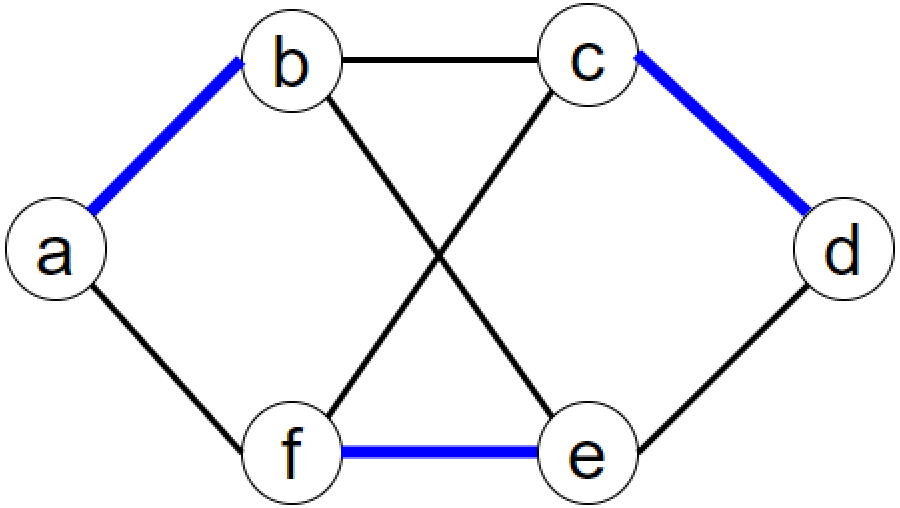
Theorem. Matching M is maximal for G iff G has no augmenting path for M.

Proof

$\Rightarrow$ Let M be a maximal matching. Suppose, G has an augmenting path π. Let us build a new matching M' such that it coincides with M beyond π and is a complement to M on edges of π.

Matchings





Due to the previous statement, $|M'| = |M| + 1$. So, we have a contradiction, and M is not a maximal matching.

Matchings

$\Leftarrow$ Let G has no augmenting path for M.

Let M^*be a maximal matching.

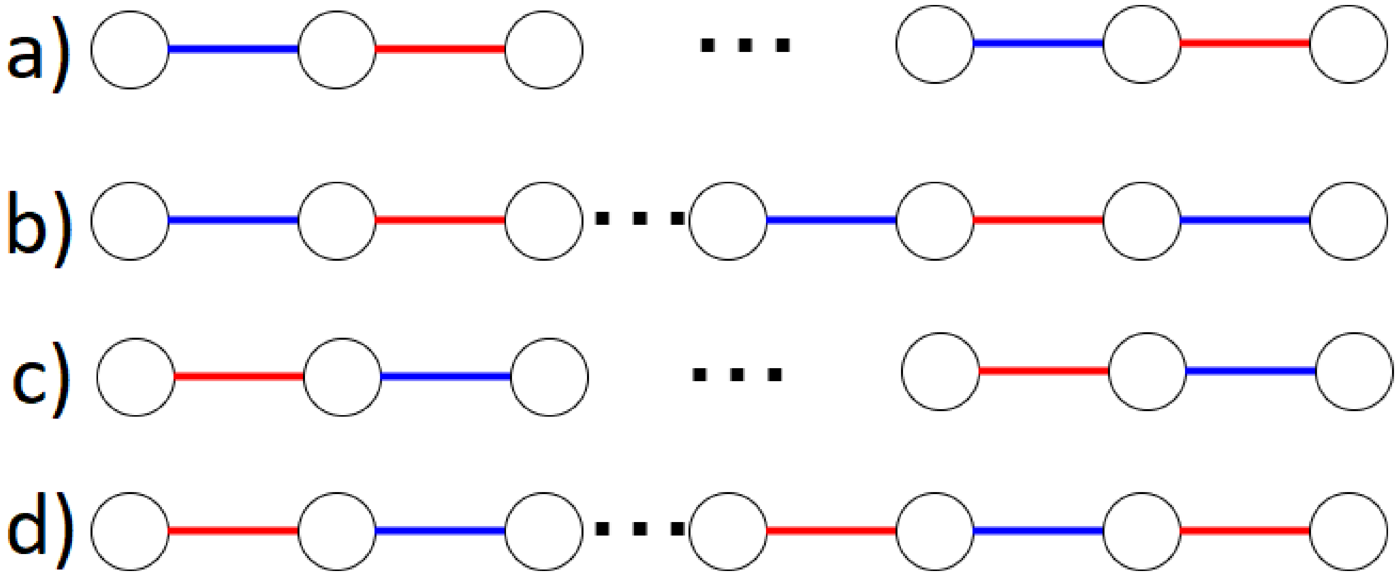
Let us consider G' which is the partial graph made by the edge set $M \Delta M^*$, where Δ stands for the symmetric difference ($M \Delta M^* = (M \cup M^*) \backslash (M \cap M^*)$).

Each vertex on G' has degree not greater than 2. Thus, each connected component of G' is either an isolated vertex or a path or a cycle.

It is evident that G' cannot contain a cycle with odd number of edges, because either M or M^* would not be a matching.

Matchings

Let us consider possible types of paths on G'.



Path of type (b) is augmenting for M, path of type (d) is augmenting for M^*. Thus, G' cannot contain such paths.

Matchings

Thus, G' is composed of alternating cycles of even length and alternating paths of even length. This implies $|M| = |M^*|$.

Thus, M is a maximal matching.

Q.E.D.

The part «$\Rightarrow$» of the proof contains description of the procedure for building a new matching with an augmenting path, and the new matching has 1 edge greater than the current matching.

Matchings

Edmonds' algorithm for building a maximal matching

Idea of the algorithm:

1) Build the initial matching M. We can use some kind of greedy algorithm to build such matching.
2) Iteratively increase the size of the current matching while this is possible:
   - Build an augmenting path.
   - Use this augmenting path to increase the matching.

How can we build an augmenting path?

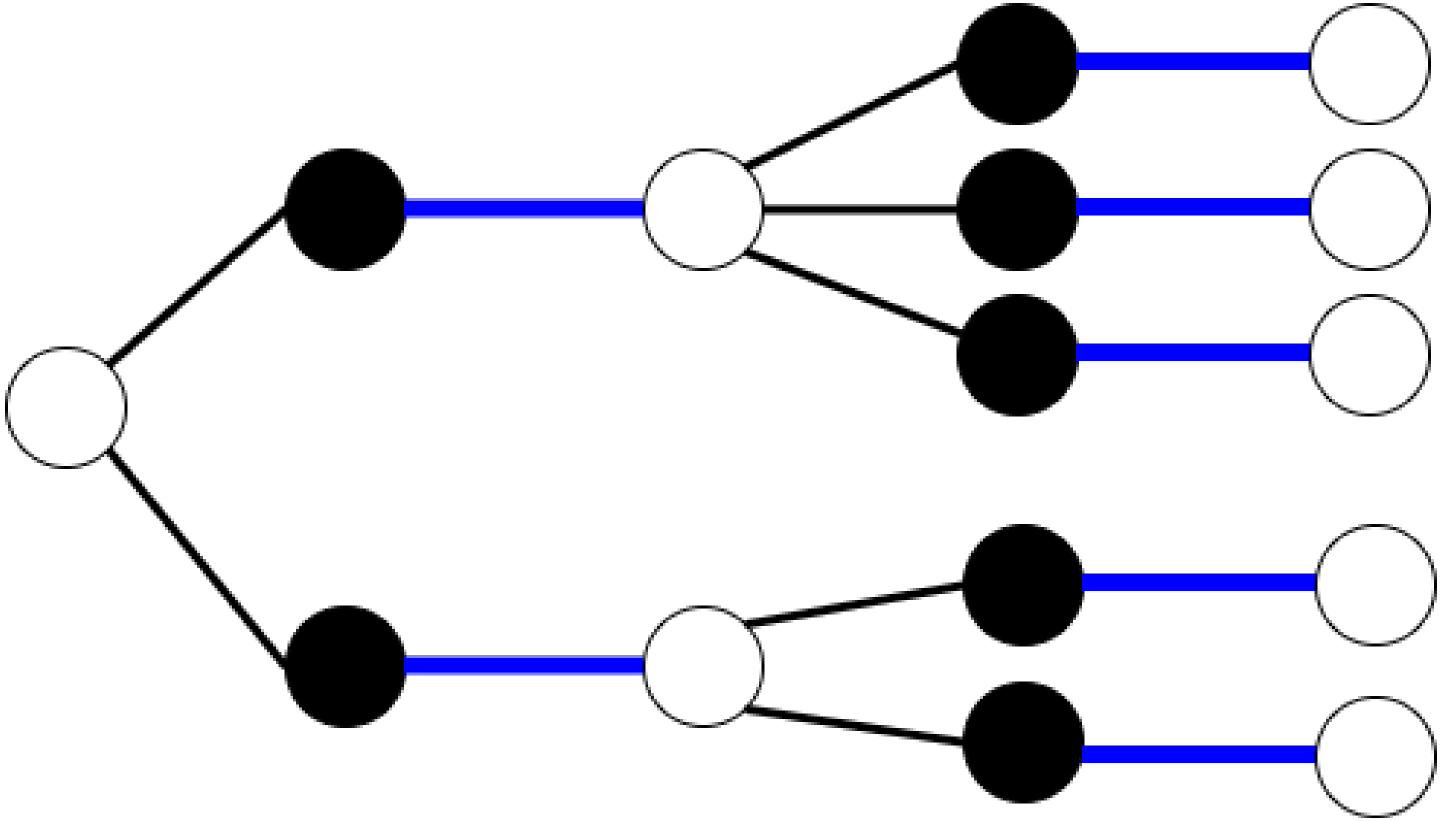
Matchings

To build an augmenting path, we can use alternating / augmenting paths and trees.

Building an alternating tree:

1) Select an arbitrary ***unmatched*** vertex as a starting vertex. This vertex becomes the root of the tree and is classified as an ***outer*** vertex.
2) Expand the tree by adding new edges to the leaves, following the rule:
   a) If the leaf is an outer vertex, add all incident unmatched edges and their endpoints (if they are not in the tree yet). The newly added vertices become ***inner*** vertices.
   b) If the leaf is an inner vertex and it is matched, add the incident matched edge and the endpoint of this edge. The newly added vertex becomes an ***outer*** vertex.

Matchings



Black vertices are inner and while vertices are outer. Matched edges are shown in blue.

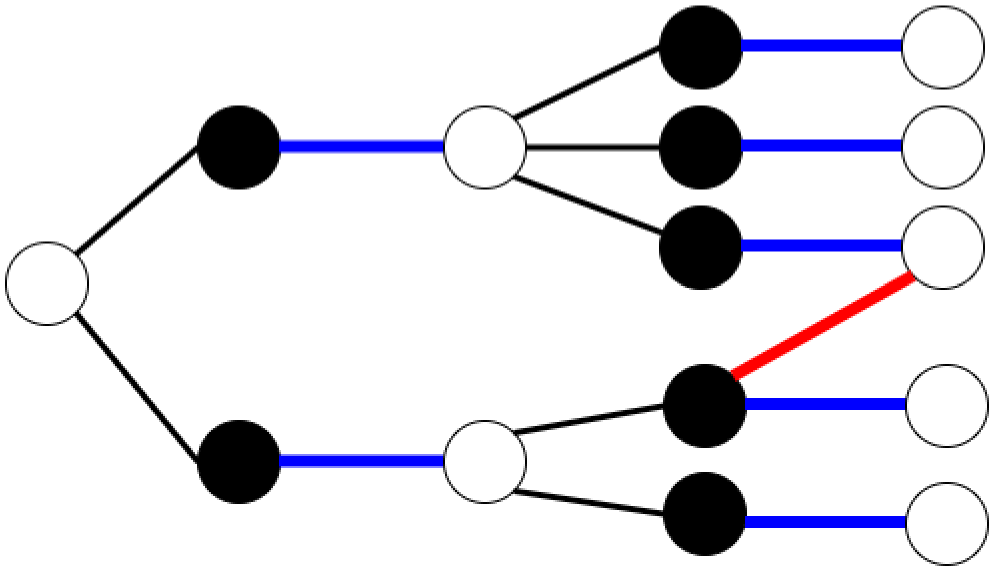
Matchings

Within this procedure the following situations can occur:

1) We added the tree an inner vertex which is unmatched. In this case, an augmenting path is found; this path starts at the tree root and finished at this unmatched vertex.

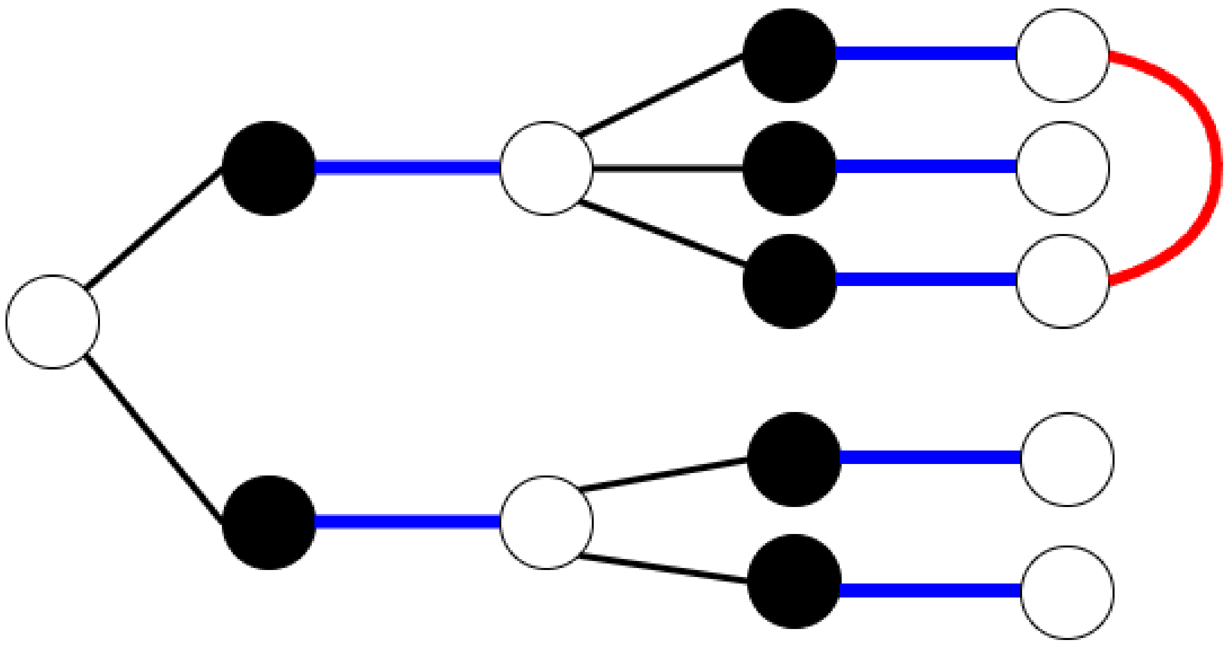
Matchings

2) At step (2a) we found an edge connecting the current outer vertex with an inner vertex. In this case, an even length cycle is found. Continue the procedure.



Matchings

2) At step (2a) we found an edge connecting two outer vertices of the tree. In this case, an odd length cycle is found. (This case cannot occur for bipartite graphs!)



Matchings

To be more precise, a ***blossom*** is found. A ***blossom*** is a cycle of odd $(2k+1)$ length with k inner vertices and $k+1$ outer vertices and one of the outer vertices is connected to the alternating tree's root with an alternating path.

In this case, we should ***shrink the blossom***. This operation means that we contract all vertices of the blossom into a new ***quasivertex*** v'; all vertices of the graph that were adjacent to the blossom's vertices, become neighbours of the new vertex v'. Shrinking the blossom results in the new graph G' and the new matching M' on it.

Theorem. There is an augmenting path for M on G $\Leftrightarrow$ there is an augmenting path for M' on G'.

(without proof).

Matchings

4) The algorithm cannot expand the alternating tree any longer. This means that all leaves of the tree are outer vertices adjacent to the inner vertices of the tree. But not all vertices of the graph are in the tree. Such tree is called *Hungarian*. Такое дерево называется *венгерским*.

In this case, select a new unmatched vertex as a root of the new alternating tree and run the procedure again.

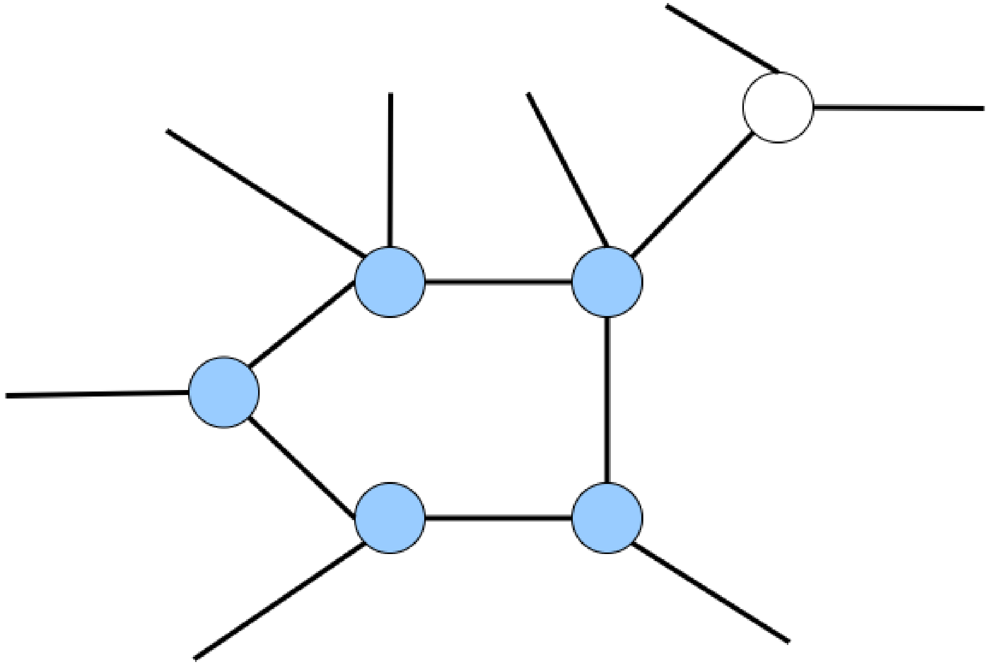
Matchings

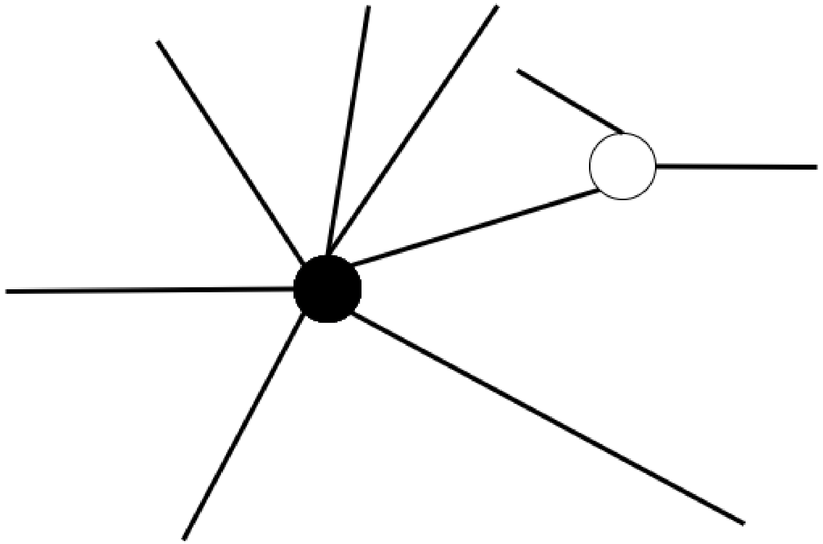
In the end of the algorithm, either an augmenting path is found, or the graph is split into a set of alternating trees connected with edges. In the latter case, the current matching is maximal.

If some blossoms were shrinked, we must ***expand*** them to restore the initial graph and build the maximal matching on it.

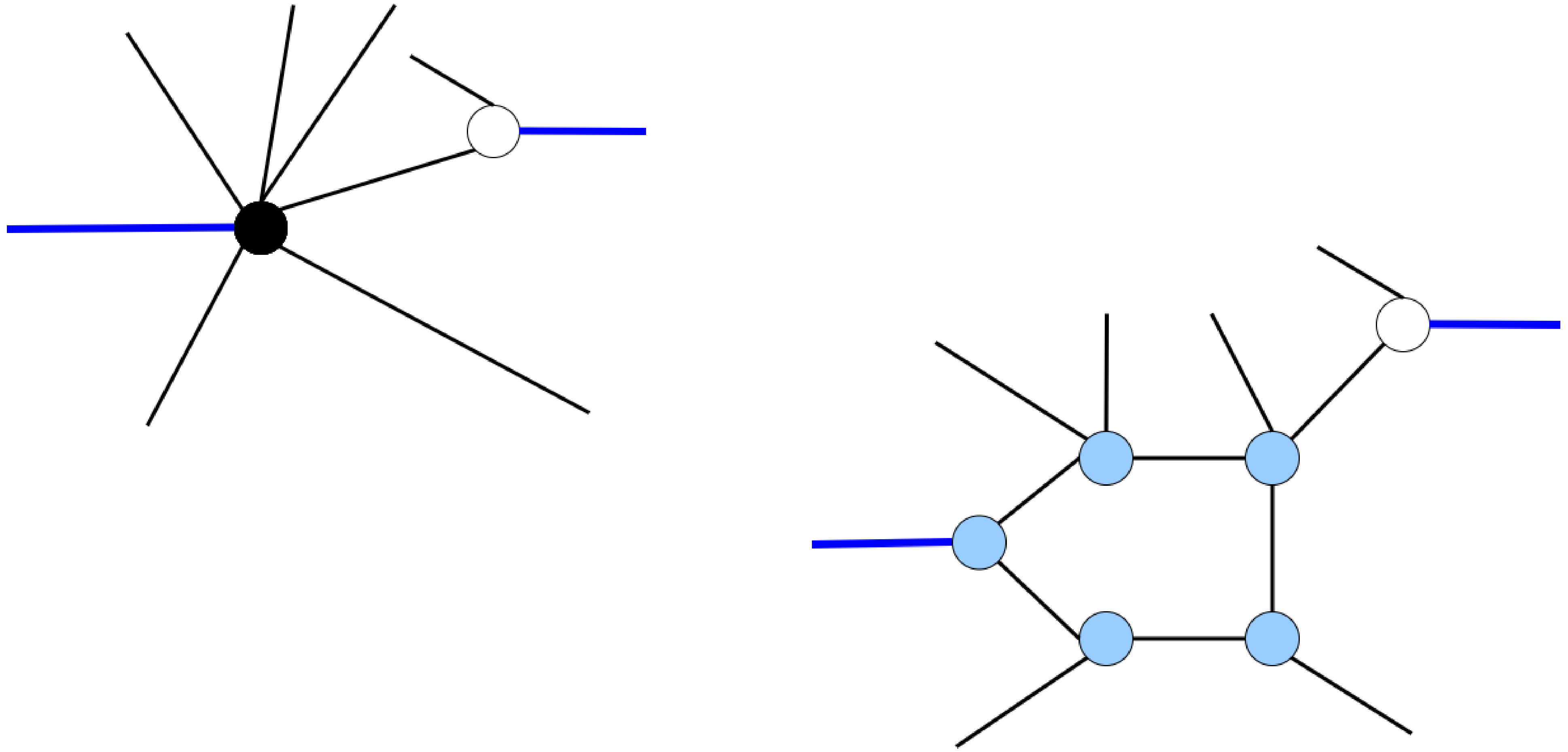
To expand a blossom, we replace the quasivertex with an initial cycle. Since the cycle has odd $(2k + 1)$ length, we always can add k edges of the cycle to the current matching to build a valid augmented matching for the initial graph.

Matchings

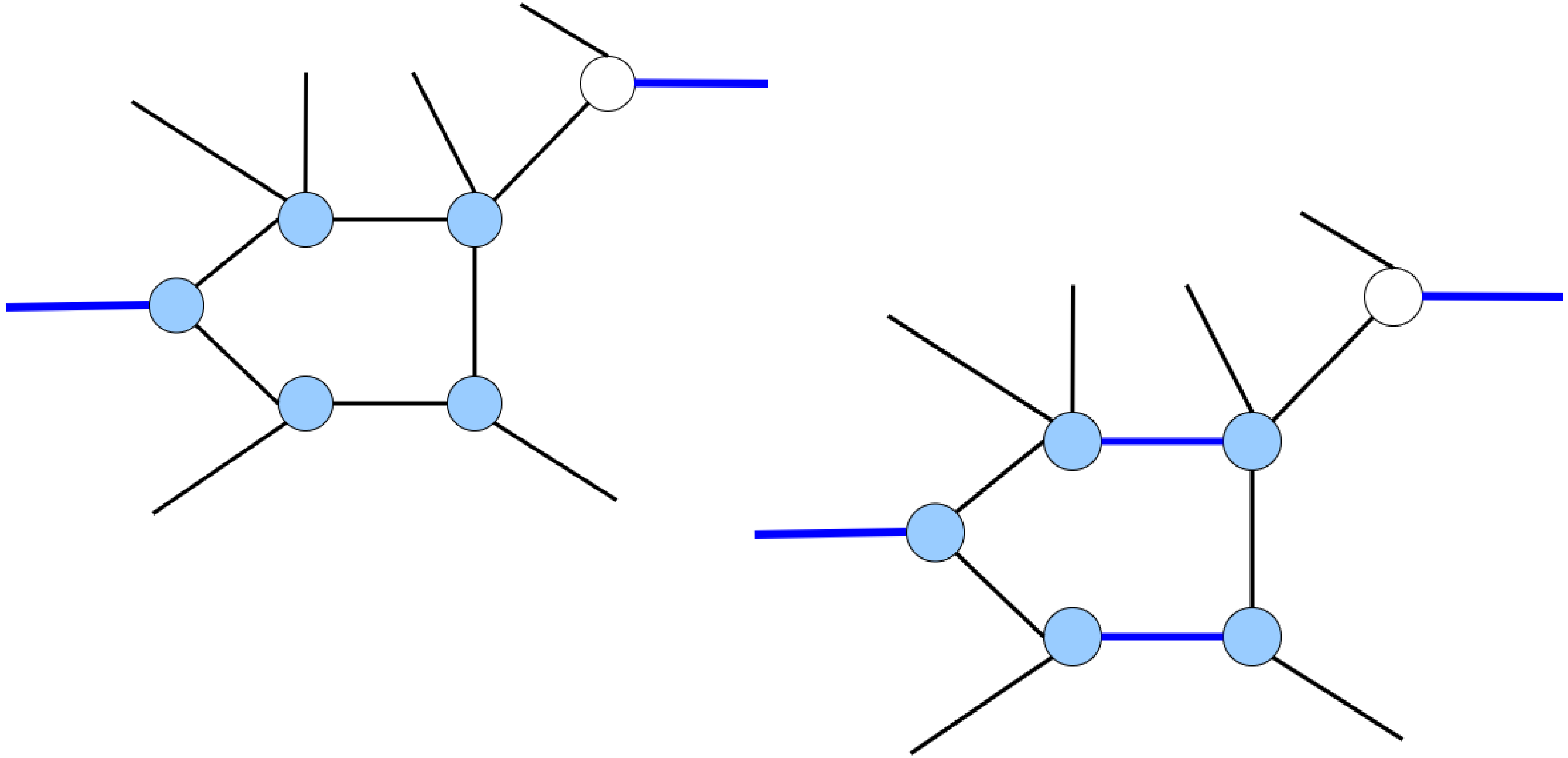




Matchings



Matchings



Matchings

Let us evaluate the time complexity of the algorithm.

- To find an augmenting path, without considering blossom shrinking, takes time $O(m) = O(n^2)$.
- A naïve implementation of shrinking one blossom takes time $O(m) = O(n^2)$. The number of recurrent shrinkings can be as much as $O(n)$.
- Thus, the overall time complexity of finding one augmenting path is $O(n^3)$.
- We can need up to $O(n)$ of iterations of finding an augmenting path + applying the found path to augment the current matching.

Thus, the overall time complexity of the algorithm is $O(n^4)$.

Matchings

If we use a smarter implementation of the blossom shrinking (keep and process the differences between G' and G, not the whole G'), the overall time complexity of the Edmonds' algorithm is $O(n^3)$.

The fastest algorithm for building maximal matching is an improved version of the Edmonds' algorithm and has time complexity $O(\sqrt{n} \cdot m)$.