Exercise 1.

1. The *neighbourhood* $N(S)$ of a set of vertices S is a set of vertices, attached to any vertex $s \in S$. For a bipartite graph $G = (V_1 \cup V_2, E)$, the sets of vertices respect the property of bipartition when talking about Hall's Condition:

$$|N(S)| \geq |S| \quad S \subseteq V$$

The following statement is to prove:

$$|N(S)| \geq |S| \forall (S \subseteq V_1 \vee S \subseteq V_2) \Leftrightarrow G \text{ is perfectly matchable}$$

Proof. Since this is an equivalency, we have to argue for both directions of implication.

$\Rightarrow$ Hall's Condition holds for any subset of V_1 or V_2. Then, $|V_1| = |V_2|$. Otherwise, it would violate Hall's Condition since $N(V_1) = V_2$ and vice versa. Suppose, that there was a maximal but no perfect matching for G. Then, there is at least one unmatched vertex in V_1 and in V_2, since $|V_1| = |V_2|$. Suppose, this unmatched vertex v is of degree 0 but then, Hall's Condition is violated since it has no neighbours. Every neighbour of v is matched since the matching was maximal and v is unmatched. Consider the alternating path starting from v and since the matching was maximal, this path can not be augmented. Let $S_1 \subseteq V_1, S_2 \subseteq V_2$ be the vertices lying on the alternating paths. Then, $|S_1 - S_2| = 1$ since v lies in either V_1 or V_2 and the rest is matched. Let without loss of generality $v \in V_1$, S_2 be the smaller set. Then $N(S_1) \subseteq S_2 < S_1$, contradicting Hall's Condition.

$\Leftarrow$ Let M be a perfect matching for G. Consider any subset S of V_1 or V_2. Then, the amount of neighbours of S is at least as large as S since there is a perfect matching, a match for every vertex of S.

□

2. For a bipartite graph $G = (V \cup W, E)$, the following holds:

$$d(v) \geq 1 \wedge d(v) \geq d(w) \forall v \in V, w \in W, (u, w) \in E \tag{1}$$

Then, there exists a matching covering all vertices of V.

Proof. Suppose that $|W| < |V|$. Then, since every vertex from V has at least an edge to W, one vertex of W must violate statement 1. So, as a consequence, $|W| \geq |V|$. Let $v \in V$. Look at $N(v), N(v) \subset W$. In fact, $|N(V_i)| \geq |V_i| \; \forall V_i \subseteq V$ for the same reason as $|W| \geq |V|$.
Match v with the neighbour of the lowest degree and eradicate v and w along with all adjacent edges. Then, $|N(v')| \geq 1$ and statement 1 still holds for the residual graph. Continue until every vertex of V is matched.

□

Exercise 2.

It is obvious that for $|M_i| \geq \frac{i}{i+1}|M^*|$ to hold for $i = 1$, $|\mathcal{P}_1| \geq \frac{1}{2}|M^*|$ needs to be true. To find a maximum set $\mathcal{P}_1$ of vertex-disjoint paths of length 1, we can use the following algorithm.

Algorithm 1: Find $\mathcal{P}_1$

Result: Find a maximum set of vertex-disjoint paths of length 1 in G
$\mathcal{P}_1 = \emptyset$;
while $E \neq \emptyset$ **do**
  $(u, v) \in E$;
  $\mathcal{P}_1 = \mathcal{P}_1 \cup (u, v)$;
  remove all edges adjacent to u or v from E;
end

For each iteration of the while loop, either a edge that is in M^*, or a edge out side of M^* will be picked. We will now look at these two cases:

1. If (u, v) is in M^*, then $|\mathcal{P}_1|$ is increased by 1. The edges that are removed from E are (u, v) plus a number of other edges which cannot be in M^*, since (u, v) is in M^*. If, for all iterations of the while loop, we pick edges that are in M^*, we therefore get $|\mathcal{P}_1| = |M^*|$.

2. If (u, v) is not in M^*, then we also increase $|\mathcal{P}_1|$ by one, but since (u, v) is not in M^*, do remove 1 or 2 edges that are in M^* (since either u or v could be free). In the second case we remove matched edge more than we add to $\mathcal{P}_1$ and therefore $|\mathcal{P}_1| \geq \frac{1}{2}|M^*|$.

For both cases, $|\mathcal{P}_1| \geq \frac{1}{2}|M^*|$ holds.

Since $|M_i| = |M_{i-2}| + |\mathcal{P}_i|$ we can show that $|M_i| \geq \frac{i}{i+1}|M^*|$ holds for $i > 1$ if

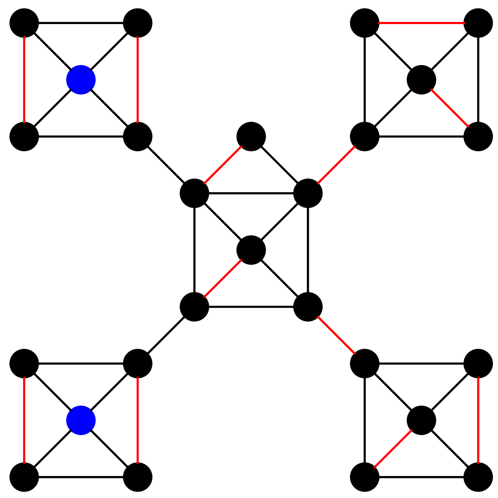
$$|\mathcal{P}_i| \geq \frac{i}{i+1}|M^*| - \frac{i-2}{i-2+1}|M^*| = \frac{2}{i^2-1}|M^*|$$

To find a maximal, vertex-disjoint set $\mathcal{P}_i$ on G with respect to M_i, each time we find a path p, which belongs to $\mathcal{P}_i$, we remove all vertices along p and their adjacent edges from G to form a reduced graph G'. We then continue to find another path in this reduced graph and repeat the above step. This ensures that the found paths are vertex-disjoint.

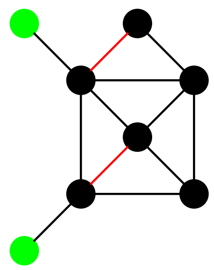
For each path p with length i which we therefore cut a segment from G which contains $i + 1$ vertices. Within these cut vertices, between 0 and $\frac{i+1}{2}$ can be free with respect to M^*. This means that for each path p we find, we remove between $i + 2$ and $\frac{i+2}{2}$ edges from G, which are in M^*.

Exercise 3.

Proof. We start with the following matching M on the graph G, where the edges in the matching are marked red.



This matching is of size 12. From the lecture we know that a matching is maximum if there are no aufmenting paths in the graph, regarding the given matching. With the given matching, there are only two vertices left unmatched (marked blue), which means there is only one possibility where we could find an augmenting path. This path would need to connect the two free vertices and therefore pass through the middle part of the graph. We know focus on the middle part to proof that we cannot find an alternating path between the two green marked vertices.



It is quite easy to see that there is no alternating path between the two green vertices, since no single edge that is not in the matching connects the ends of the two edges in the matching. Therefore the graph with matching M does not contain a augmenting path and M of size 12 is maximum.

□

Exercise 4. *Imperfect edges*

a)

b) As already mentioned in the notes, we are able to find a perfect matching in $\mathcal{O}(n^{2.5})$. Let $G' = (G, M)$ be the graph for a given graph and a given matching M. G' contains of all vertices and all the edges which are not part of the matching. If there is a cycle C in G', then $M' = M \triangle E_C$ is still a perfect matching. The symmetric difference removes the already matched edges in M und adds the remaining edges of C, think of alternating paths. The polytime algorithm works as follows:

1. Result $= E$
2. Calculate a perfect matching in $\mathcal{O}(n^{2.5})$
3. $G' = (G, M)$
4. Calculate cycles in G' and save the edges in $\mathcal{O}(n + m)$
5. $M' = M \triangle E_C$ for a cycle C
6. $G' = (G', M')$ (alter the graph)

The remaining edges in Result were never part of a perfect matching, therefore imperfect.