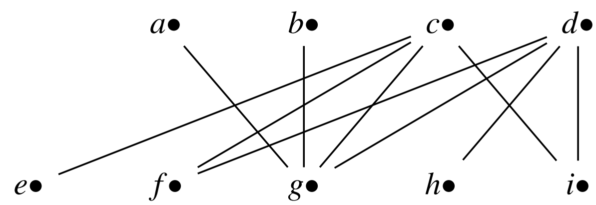
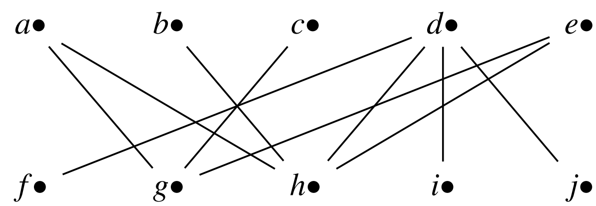
1. 3.1.1. Find a maximum matching in each graph below. Prove that it is a maximum matching by exhibiting an optimal solution to the dual problem (minimum vertex cover). Explain why this proves that the matching is optimal.

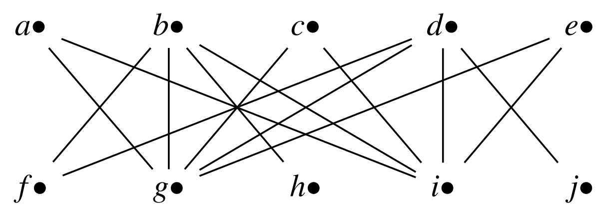
a)



b)

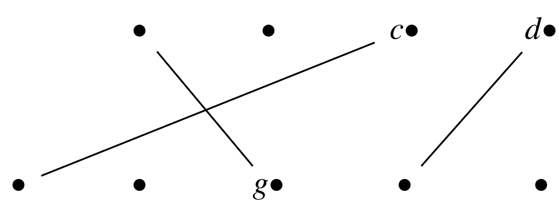


c)



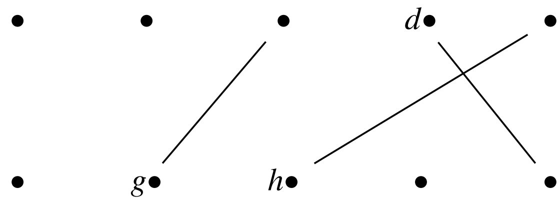
Solution:
Each of the graphs above are bipartite. Hence, by the Konig Egavary Theorem, the amount of edges in a maximum matching is the same size as the number of vertices in a minimum vertex cover. The following solutions are all optimal because the cardinality of each vertex cover is the same as the cardinality of each matching.

Solution, part a:
Minimum vertex cover has 3 vertices, such as c, d, g. Example of a maximum matching on those vertices:

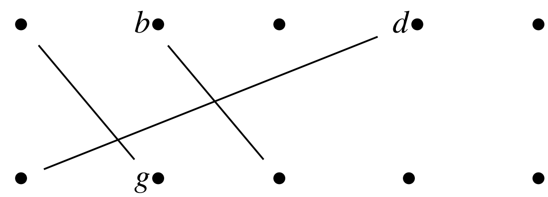


Solution, part b:

Minimum vertex cover has 3 vertices, such as d, g, h. Example of a maximum matching:

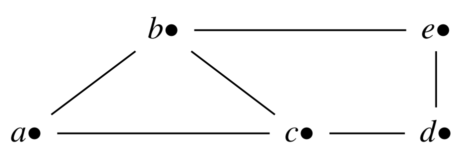


Solution, part c:
Minimum vertex cover has 3 vertices, such as b, d, g. Example of a maximum matching on those vertices:



2. 3.1.3. Let S be the set of vertices saturated by a matching M in a graph G. Prove that some maximum matching also saturates all of S. Must the statement be true for every maximum matching?

Solution:
If M is maximum, then we are done. Otherwise, suppose M is not maximum. By Theorem 3.1.10., this implies that G has an M-augmenting path p. Let M' be a matching including all of M except that M' includes the maximum matching on p. Then $V(M') \supset V(M)$. Hence, if M' is a maximum matching, we are done. Otherwise, apply Theorem 3.1.10. again and apply the same argument to find a larger matching M'' such that $V(M'') \supset V(M') \supset V(M)$. This algorithm always produces a matching that saturates S. It will end when a maximum matching is found. The statement may not be true for every maximum matching. In the graph below, $S = \{a, b, c, d\}$. Consider matching $M = \{ab, cd\}$. M is a maximum matching. Yet, $M' = \{ac, be\}$ is a maximum matching that does not saturate all of S, namely M' does not saturate e.



□

3. 3.1.19. Let $\mathbf{A} = (A_1, \ldots, A_m)$ be a collection of subsets of a set Y. A **system of distinct representatives** (SDR) for $\mathbf{A}$ is a set of distinct elements $a_1, \ldots, a_m$ in Y such that $a_i \in A_i$. Prove that $\mathbf{A}$ has an SDR if and only if $|\cup_{i \in S} A_i| \geq |S|$ for every $S \subseteq \{1, \ldots, m\}$. (Hint: Transform this to a graph problem.)

Solution:
We can represent the SDR by an $\mathbf{A}$-Y bipartite graph such that for all $A_i \in \mathbf{A}$, $A_i \leftrightarrow y_j, y_j \in Y$ if and only if $y_j \in A_i$. Hence, by Hall's Matching Theorem, $\mathbf{A}$ has an SDR if and only if $|\cup_{i\in S} A_i|$ for every $R \subseteq X$, $|R| \leq |N(R)|$.Observe,

$$\begin{aligned} \forall R \subseteq X, |R| \leq |N(R)| &\leftrightarrow \forall R \subseteq X, |R| \leq |\cup_{i\in R} A_i| \\ &\leftrightarrow \forall S \subseteq [m], |S| = |R| \leq |\cup i \in S A_i| \end{aligned}$$

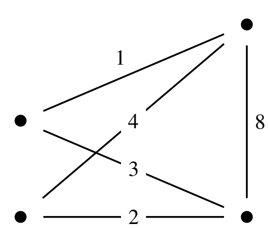
□

4. 3.1.21. Let G be an X,Y-bigraph such that $|N(S)| > |S|$ whenever $\emptyset \neq S \subset X$. Prove that every edge of G belongs to some matching that saturates X.

Solution:
Let $xy \in E(G)$. Consider $G' = G - x - y$. Let $S' \subset V(G') \cap X$. Then $S' \subset X$. By assumption $|S| < |N_G(S)|$. Thus, $|N'_G(S')| \geq |S'|$. Hence, there exists a matching in G' saturating $X - x$, which means G has a matching saturating X using the edge xy. □

5. 3.2.1. Using nonnegative edge weights, construct a 4-vertex weighted graph in which the matching of maximum weight is not a matching of maximum size.

Solution:
In the graph below, the matching including only the edge with weight 8 has maximum weight, but there are larger matchings (a matching including the edges with weights 4 and 3, for example).



6. 3.2.5a. Find a transversal of maximum total sum (weight) in the matrix below. Prove that there is no larger weight transversal by exhibiting a solution to the dual problem. Explain why this proves that there is no larger transversal.

$$A = \begin{pmatrix} 4 & 4 & 4 & 3 & 6 \\ 1 & 1 & 4 & 3 & 4 \\ 1 & 4 & 5 & 3 & 5 \\ 5 & 6 & 4 & 7 & 9 \\ 5 & 3 & 6 & 8 & 3 \end{pmatrix}$$

Solution:

The Hungarian Algorithm applied to this problem produces:

$$\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 1 & 2 \\
\hline
4 & \underline{0} & 1 & 2 & 2 & 0 \\
2 & 1 & 2 & \underline{0} & 0 & 0 \\
3 & 2 & \underline{0} & 0 & 1 & 0 \\
7 & 2 & 2 & 5 & 1 & \underline{0} \\
7 & 2 & 5 & 3 & \underline{0} & 6
\end{array}$$

Hence, the sum of the maximum weight transversal is

$$a_{11} + a_{23} + a_{32} + a_{45} + a_{54} = 9 + 8 + 4 + 4 + 4 = 29.$$

We also find the sum of the minimum cover by summing up the values on the edge of the output of the algorithm:

$$4 + 2 + 3 + 7 + 7 + 0 + 1 + 2 + 1 + 2 = 29.$$

Since the sum of the minimum cover is equal to the sum of the maximum weight transversal, by Lemma 3.2.7. the solution must be optimal. □

7. 3.2.7. *The Bus Driver Problem.* Let there be n bus drivers, n morning routes with durations $x_1, \ldots, x_n$, and n afternoon routes with durations $y_1, \ldots, y_n$. A driver is paid overtime when the morning route and afternoon route exceed total time t. The objective is to assign one morning run and one afternoon run to each driver to minimize the total amount of overtime. Express this as a weighted matching problem. Prove that giving the ith longest morning route and ith shortest afternoon route to the same driver, for each i, yields an optimal solution. (Hint: Do not use the Hungarian Algorithm; consider the special structure of the matrix.) (R.B. Potts)

Solution:
Let each route be a vertex. The duration of a route is the weight of its vertex. Since we are only interested in the amount of overtime, we represent this problem using a matrix of edge weights minus t. Consider the matrix below where the x_i are in non-increasing order and each entry a_{ii}, is the amount of overtime driver i has.

$$\begin{array}{cccccc}
 & y_1 & y_2 & y_3 & \ldots & y_n \\
x_1 & \max\{x_1 + y_1 - t, 0\} & & & & \\
x_2 & & \max\{x_2 + y_2 - t, 0\} & & & \\
x_3 & & & \max\{x_3 + y_3 - t, 0\} & & \\
\ldots & & & & & \\
x_n & & & & & \max\{x_n + y_n - t, 0\}
\end{array}$$

Since we set the order of the x_i, the goal is to arrange the y_i so that for every x_i and y_i,

$$a_{ii} = \max\{x_i + y_i - t, 0\}$$

is minimized. Whenever $x_i + y_i \leq t$, $a_{ii} = 0$. The goal is for each a_{ii} to be as close to 0 as possible. We iteratively choose a partner for each x_i, starting with the largest, x_1, because this route has the potential for the largest amount of overtime. The optimal choice is the smallest y_i, call it y_{s_1}. We now move on to the second largest x_i, x_2. The same rule applies: the best choice is the smallest y_i. Yet, since we already used y_{s_1}, the best we can do is the second smallest, y_{s_2}. This continues until we reach x_n, at which point the only y_i left will be the largest one. Hence, giving the ith longest morning route and the ith shortest afternoon route to the same driver, for each i, yields an optimal solution. $\square$