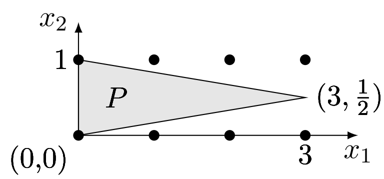
Caleb Logemann
MATH 566 Discrete Optimization
Homework 10

1. Solve the following problem using branch and bound. Draw the branching tree too.

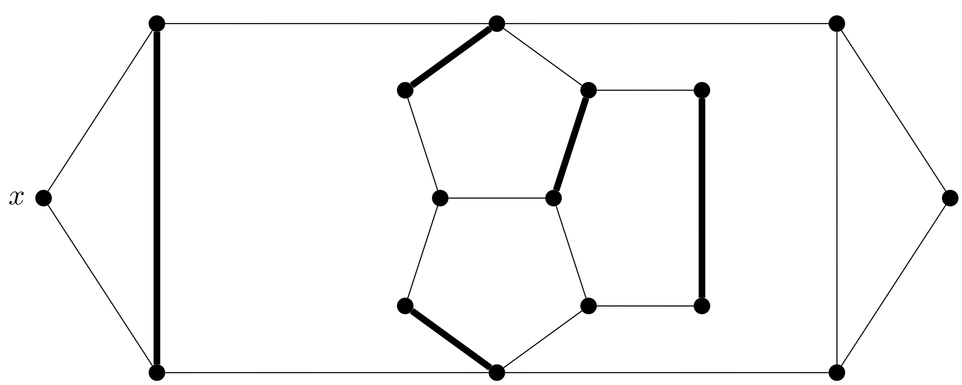
$$(P) = \begin{cases} \text{maximize} & -x_1 + 4x_2 \\ \text{subject to} & -10x_1 + 20x_2 \leq 22 \\ & 5x_1 + 10x_2 \leq 49 \\ & x_1 \leq 5 \\ & x_i \geq 0, x_i \in \mathbb{Z} \text{ for } i \in \{1, 2\} \end{cases}$$

   You can use any linear programming solver for solving the relaxations.

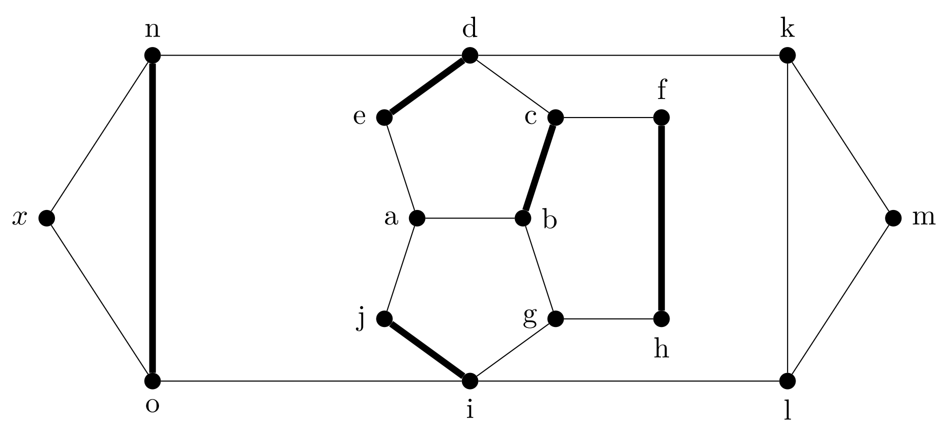
2. Let P be a convex hull of $(0,0), (0,1), (k, \frac{1}{2})$. Give an upper bound on Chvátal's rank of P. (Show it is at most $2k$, actually, it is exactly $2k$.)
   Drawing of P for $k = 3$.



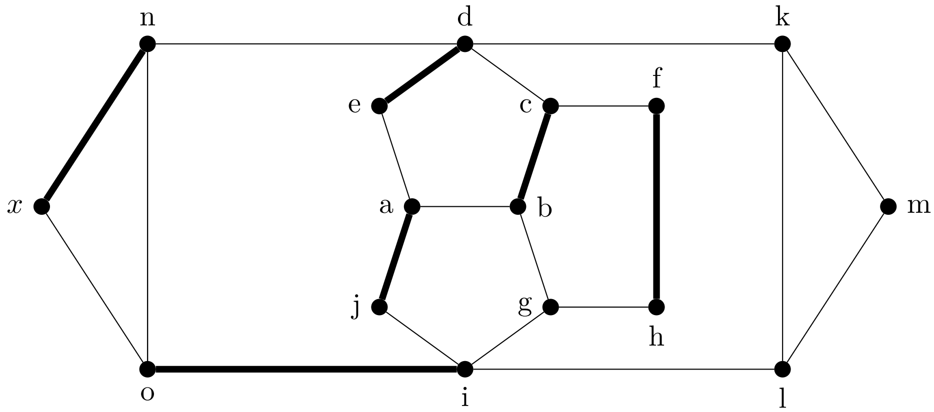
3. Run Edmond's Blossom algorithm on the following graph. Notice that somebody already found a partial matching. What is the largest possible matching? Try to start growing augmenting tree from x, use BFS algorithm for building the tree.



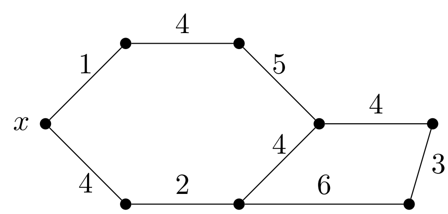
   First I will label all of the vertices of the graph as follows.



The first step in the algorithm is to grow forest from the exposed vertices. The following is one possible forest that could be grown into an augmenting path.

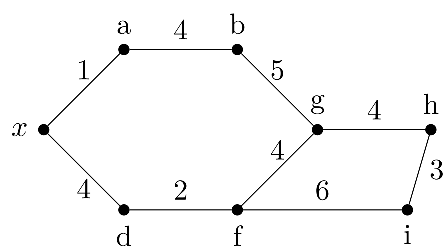


4. Find minimum-weight perfect matching in the following graph:



(a) By using algorithm from class that grows augmenting tree (and keep primal/dual solutions). Start growing x.

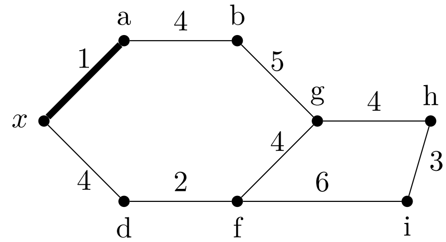
First I will relabel the vertices of $G = (V, E)$ as follows.



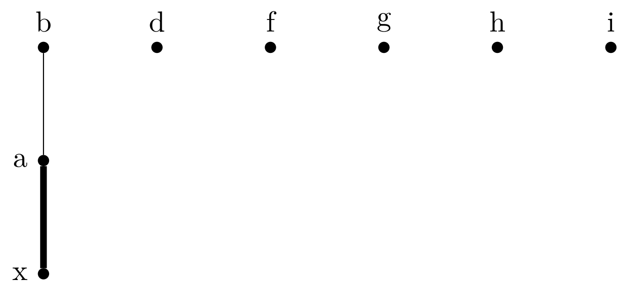
Initially we will start with initial solutions $y_v = 0$ for all $v \in V$ and $x_e = 0$ for all $e \in E$. The initial solution $\mathbf{y}$ is a feasible solution to the dual problem as $y_u + y_v <= c(e_{uv})$ for all edges. In this case $E_= = \{\}$ because there are no edges $e = (u, v)$ such that $y_u + y_v = c(e)$. Now we will construct the initial alternating forest using $E_=$. Since $E_=$ is empty the alternating forest will contain all vertices as single node trees.

Since a perfect matching is not possible, we must modify $\mathbf{y}$ for one of the connected components of F, the alternating forest. Therefore we will change y_x. The smallest possible increase in y_x that maintains feasibility of the dual is one, so let $y_x = 1$. Now $E_= = \{(x, a)\}$. We can now start growing the alternating forest again with edges from $E_=$.

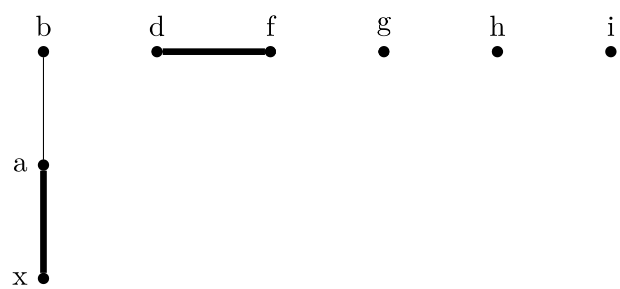
The only augmenting path is from x to a. The matching that is created is shown below with bold edges.



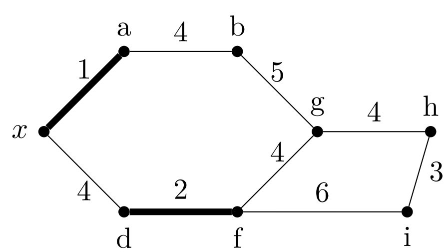
This does not creat a perfect matching so we must change a value of $\mathbf{y}$ to allow one of the connected components of F to be expanded. Consider the vertex b, y_b can be increased by 4 so that $y_b = 4$ and $E_= = \{(x,a),(a,b)\}$. Now the alternating forest can be grown again into a matching. If we first match x and a as before, then the final forest would look like



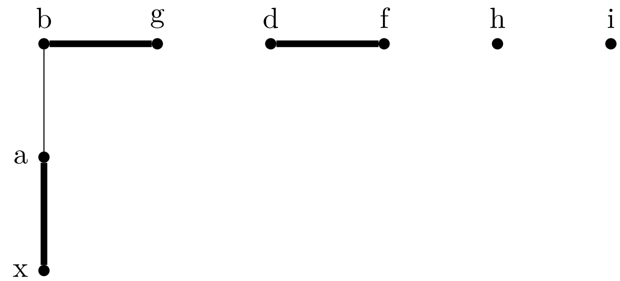
This does not find a perfect matching. In fact this is the same matching as before. Therefore $\mathbf{y}$ must be modified for one of the connected components of F. Consider modifying y_d. Let $y_d = 2$ so that (d,f) can be added to $E_=$. Now $E_= = \{(x,a),(a,b),(d,f)\}$. Building on the previous alternating forest we can find an alternating path from d to f



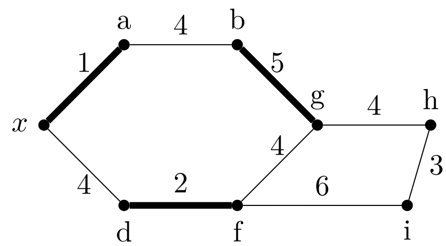
This forest creates the matching shown below.



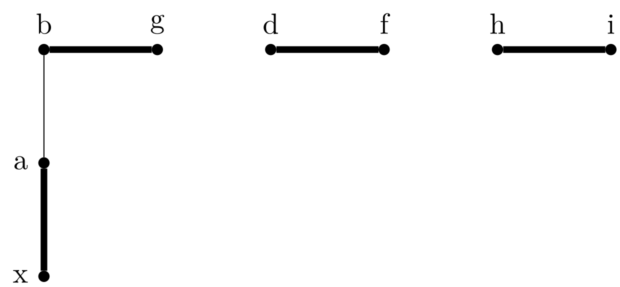
This matching isn't perfect, therefore $\mathbf{y}$ needs to be updated to allow for the expansion of F. The value of y_g can be set to 1, to allow (b,g) to be added to $E_=$. This allows for the alternating forest to be expanded as follows.



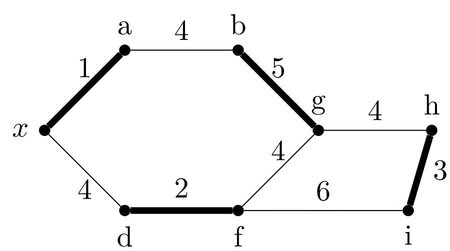
The matching from this alternating forest is shown below.



Again this is not a perfect matching, therefore y_h must be set to 3. This adds the edges (g, h) and (h, i) to $E_=$. The set $E_=$ is now $\{(x, a), (a, b), (d, f), (b, g), (g, h), (h, i)\}$. Growing an alternating forest from this set of edges allows us to find an alternating path from h to i. The final alternating forest is shown below, with no exposed vertices.



Since all edges are covered this is a perfect matching. Now since we have a feasible solution to both the primal and the dual problem, this is the optimal solution to the minimal matching problem. This matching is shown below.



Also to recap here are the values of $\mathbf{y}$.

$$\begin{aligned}
y_x &= 1 \\
y_a &= 0 \\
y_b &= 4 \\
y_d &= 2 \\
y_f &= 0 \\
y_g &= 1 \\
y_h &= 3 \\
y_i &= 0
\end{aligned}$$

(b) Formulate the problem using Integer/Linear programming and solve it with your favorite solver.

5. Slither is a two-person game played on a graph $G = (V, E)$. The players, called First and Second, play alternatively, with First playing first. At each step the player whose turn it is chooses a previously unchosen edge. The only rule is that at every step the set of chosen edges forms a path. The loser is the first player unable to make a legal move at his or her turn. Prove that if G has a perfect matching, then First can force a win.

6. Implement algorithm for finding maximum matching in bipartite graphs. Test it on the 3D-cube.

7. Implement algorithm for finding maximum matching in any graph.
   Test it on the 3D-cube and the graph from question 3.
   (Doing this will also solve the previous question - 2 for 1.)