Please enter your name and uID below.

Name:
uID:

Submission notes

- Due at 11:59 pm on Friday, October 23.
- Solutions must be typeset using one of the template files. For each problem, your answer must fit in the space provided (e.g. not spill onto the next page) *without* space-saving tricks like font/margin/line spacing changes.
- Upload a PDF version of your completed problem set to Gradescope.
- Teaching staff reserve the right to request original source/tex files during the grading process, so please retain these until an assignment has been returned.
- Please remember that for problem sets, collaboration with other students must be limited to a high-level discussion of solution strategies. If you do collaborate with other students in this way, you must identify the students and describe the nature of the collaboration. You are not allowed to create a group solution, and all work that you hand in must be written in your own words. Do not base your solution on any other written solution, regardless of the source.

1. (Choosing Candy) [15 points] A SoC faculty member is going to the store to buy Halloween candy. Since they're faculty, this trip happens at the last minute, so unfortunately the selection is abysmal. Only two (terrible) kinds of candy remain - Necco Wafers, which cost \$3 for a bag for 30 mini-rolls and Good & Plenty, which cost \$5 for a bag of 45 mini-boxes. Regardless, candy must be obtained to avoid tricks from the trick-or-treaters. He expects 90 kids, and wants to have at least 5 pieces of candy per expected visitor. There are only 8 bags of Necco Wafers and 10 bags of Good & Plenty left on the shelf. Further, the professor is quirky, and wants to buy at least as many bags of Good & Plenty as he does Necco Wafers. What combination of candy should they buy to minimize cost?

   a. Write a linear program for solving this problem. Your LP **must be in canonical form**, and use the variable x_1 for Good & Plenty and x_2 for Necco Wafers.
   If using LaTeX, the align environment is great for displaying LPs.

   b. Plot the feasible region for your LP. Good & Plenty should be on the x-axis, and Necco Wafers should be on the y-axis. You must label each line with its corresponding constraint, and clearly indicate the feasible region. Draw and label an "isoprofit" line corresponding to spending \$60 on candy.
   Tip: You can use Desmos (desmos.com) to create this figure. In order to highlight the feasible region, you may either (a) plot the opposite of each inequality in your LP - this will shade the infeasible regions, leaving the feasible region unshaded, or (b) input your inequalities as a system, e.g. "$2x + y \geq 10$ $\{x \geq 0\}\{y \leq 3\}$", which will highlight only the region satisfying all of them. Be sure to include the equation list in your figure, otherwise the lines will be unlabelled.

   c. Provide a table that shows the value of your objective function at each corner (vertex) of the feasible region. There should be a row for each corner. The first column should give the x-coordinate (x_1 value), the second column the y-coordinate (x_2 value), and the third column the objective function value.

   d. What is the value of the optimal solution to this LP? How much will the professor spend on candy?

2. (LP Duality) [20 points] Consider the linear program from Problem 1.

   a. Argue that the constraints arising from the limited number of bags of candy on the shelf do not affect the optimal solution.

   b. Drop these constraints from your LP. What is the dual of this (smaller but equivalent) linear program? Use variables $y_1, y_2, \ldots$ and match the standard dual formulation given in Erickson. Please write out your inequalities (do not use matrix form).

   c. Plot the feasible region for the dual LP. You must label each line with its corresponding constraint, and clearly indicate the feasible region.
   See plotting tips using Desmos on Problem 1.

   d. Provide a table that shows the value of your objective function at each corner (vertex) of the feasible region.

   e. What is the value of the optimal solution to the dual LP?

3. (Complement Connectivity) [15 points]

   Recall that the *complement* of a graph $G = (V, E)$ is the graph $\bar{G} = (V, [V]^2 \setminus E)$ on the same vertex set where two vertices are adjacent if and only if they were non-adjacent in G.

   Prove or disprove that if a graph G is disconnected, its complement $\bar{G}$ is connected.

4. (Greedy Coloring, Part 1) [25 points]

   A *proper coloring* of the vertices of a graph G is an assignment of colors (typically represented by integers) to each element of $V(G)$ so that if $uv \in E(G)$, the color of u is not equal to the color of v. An optimal solution to the coloring problem is a proper coloring using the minimum possible number of colors (for any proper coloring). A greedy approach to coloring is described by the following pseudocode.

```
Algorithm 1: GreedyColoring
   Input: $G = (V, E)$ as adjacency lists $A[i]$
1  color $\leftarrow$ int[$n$]
2  S $\leftarrow$ new Set($1, \ldots, n$)
3  for $i$ from 1 to $n$ do
4      color[$i$] $\leftarrow 0$
5  end
6  for $i$ from 1 to $n$ do
7      C $\leftarrow$ new Set({color[$j$] such that $j$ in $A[i]$})
8      V $\leftarrow S \setminus C$
9      color[$i$] $= \min(V)$
10 end
11 return color
```

   a. Prove that GreedyColoring uses at most $1 + \Delta$ colors where Δ is the maximum degree in G.

   b. The timing analysis of this algorithm depends on the implementation of lines 7-9. Give more detailed pseudocode for implementing this so that the overall execution of lines 7-9 takes no more than $O(deg[i])$ operations (following pointers, array lookups, etc are single operations; you are not required to use Sets as outlined in the current algorithm).

   c. Assuming that lines 7-9 take $O(deg[i])$ steps, show that GreedyColoring is linear in the size of the input, $O(V + E)$.

5. (Greedy Coloring, Part 2) [25 points]

   a. Prove that GreedyColoring can fail dismally: for every even positive integer n, construct a bipartite graph G_n with n vertices such that the greedy coloring algorithm uses $n/2$ colors (instead of the 2 colors that would suffice).

   b. Consider modifying GreedyColoring to take in a permutation of the vertices which specifies the order of iteration for the greedy step. This new routine has signature Permuted-Greedy($G, \pi[1..n]$), and is identical to before except that line 6 now reads `for(`$j = 1$ `to` n`) do` $i \leftarrow \pi[j]$. Give an example of a connected graph on at least three vertices for which PermutedGreedy will always produce an optimal solution (regardless of the permutation selected).

   c. Does PermutedGreedy still produce a suboptimal solution on your graph from part (a) for all permutations? Argue why, or describe a situation where it does not.