Math/CS/Econ 210 Spring 2021 —HW # 3 Assignment Due Thu. Mar. 4, 5 PM

Submission Instructions: Problems 1, 2, 5, and 6 can be done with your lab partner, with one of you submiting the notebook by downloading it as an `ipynb` file and uploading that to the HW3 Moodle link. You can submit four separate notebooks or merge them into a single notebook. Also, each student should upload a file with their solution to Problems 3 and 4.

Reading: Strayer 2.5 (see also the lecture slides/recordings for Fri 2/26 through Wed 3/3)

Reminder of HW collaboration rules: I encourage you to discuss HW problems with other students in the class and/or with me. Your submission should reflect your personal understanding, so you must write the solutions yourself without referring to notes from your collaborative work. For computer problems, this means you may not cut-and-paste (or anything of that nature) except from sample code that I provide or from your own previous work.

1 (Coding): Download the notebook `vector-matrix-loop-tutorial` from Moodle (Python section); upload it to either the Jupyterhub or `colab.research.google.com`; read the top portion (tutorial on vectors, matrices, and loops); and do the three problems at the end.

2 (Coding): Download the notebook `hw3prob2` from Moodle (Python section), upload it to either the Jupyterhub or `colab.research.google.com`; read the top portion, which is a demonstration of "if statements" and "max/min" in Python.

The last three cells in `hw3prob2` provide skeletons of two Python functions `target` and `select` for you to write, followed by a cell that might be useful for testing these functions. The functions should implement the `Target` and `Candidates/Select` steps of `SimplexBF`. Specifically:

(a) Write a Python function `target` with:

Input: a tableau (just the numbers, not the labels)
Output: the column number of the largest positive c_j
(or output -1 if there is no positive c_j)

(b) Write a Python function `select` with

Inputs: a tableau (just the numbers, not the labels)
a column number (where the target is)
Output: the row number of the pivot that `SimplexBF` would choose in that column
(or output -1 if there is no positive a_{ij} in this column)

Some comments/suggestions:

• One idea for part (a): set up an empty vector (a 1-dimensional array) of the appropriate size and then fill it with the values of c_j from the tableau. Once you have that, the function `np.argmax` (as explained at the top of `hw3prob2.ipynb`) will pretty much finish the job.

• For part (b), the `Select` step is subtler than `Target`; it's not a straight max or min, since you only consider positive a_{ij}. Thus, a more "by-hand" approach (as explained at the top of `hw3prob2.ipynb`) may be easiest, though feel free to use whatever clever idea you think of.

• In each part, don't worry about ties for largest or smallest; just choose any option among those tied.

3 (Theory): There is one way that `SimplexBF` can "crash": at the `Candidates` step, all the a_{ij} above the targeted c_j obey $a_{ij} \leq 0$, so we have no candidates and cannot proceed.

Actually... that's not a failure: it indicates that the problem is unbounded. Rather than write a full proof of this fact, let's consider a specific example to get the main idea. Consider the basic-feasible Canonical Max tableau:

x	y	-1	
-1	1	1	$= -t_1$
$1/2$	$-1/2$	$1/2$	$= -t_2$
$-1/3$	1	0	$= obj$

`SimplexBF` says the first pivot is a_{12}, at which point the tableau becomes:

x	t_1	-1	
-1	1	1	$= -y$
0	$1/2$	1	$= -t_2$
$2/3$	-1	-1	$= obj$

Notice that now the positive c_j (the one in column 1) has only zero or negative a_{ij} above it.

(a) Since the tableau is basic-feasible, we know that if we set $x = 0$ and $t_1 = 0$, and then use the constraint equations to solve for y and t_2, that will be a point in the constraint set. What values of y and t_2 does that yield? What is the value of the objective function at this point?

(b) Now consider instead setting $x = M$ (for $M > 0$) and $t_1 = 0$. In terms of M, what values of y and t_2 does that yield, and what is the value of the objective function at this point? As $M \to +\infty$, verify that this point (x, y) remains in the constraint set and that the objective function goes to $+\infty$.

(c) To see how this works more generally, let's change the tableau to the form:

x	t_1	-1	
A	1	C	$= -y$
B	$1/2$	D	$= -t_2$
E	-1	-1	$= obj$

where $A, B \leq 0$ and $E > 0$ (the combination that causes `SimplexBF` to crash), and $C, D \geq 0$ (as would be true for a basic-feasible problem). Show that $x = M$ (for $M > 0$) and $t_1 = 0$ yields points (x, y) that are in the constraint set and for which the objective function goes to $+\infty$ as $M \to +\infty$.

4. (Theory) A property of `SimplexBF` is that if you start with a basic-feasible tableau, and pivot according to the rules of `SimplexBF`, the new tableau will still be basic-feasible. Let's have you prove that.

Suppose we have a general Canonical Max tableau

x_1	x_2	$\cdots$	x_n	-1
a_{11}	a_{12}	$\cdots$	a_{1n}	b_1
a_{21}	a_{22}	$\cdots$	a_{2n}	b_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_{m1}	a_{m2}	$\cdots$	a_{mn}	b_m
c_1	c_2	$\cdots$	c_n	d

such that $b_i \geq 0$ for all i.

Suppose that a_{kj} is a pivot determined following the rules of `SimplexBF`. Consider this fragment of the tableau that we would use to compute the new value of b_i for some $i < k$:

$$\begin{array}{ccc} a_{ij} & \cdots & b_i \\ \vdots & \ddots & \vdots \\ a_{kj} & \cdots & b_k \end{array}$$

(a) Compute the new value of b_i after we pivot on a_{kj}. (Let's call this new value b_i^{new}.)

(b) Show that $b_i^{new} \geq 0$ if $a_{ij} \leq 0$. (Hint: remember that $b_i, b_k \geq 0$ and remember the rules of `SimplexBF`.)

(c) Show that $b_i^{new} \geq 0$ if $a_{ij} > 0$. (Hint: express b_i^{new} so that it involves both b_i/a_{ij} and b_k/a_{kj}.)

CONTINUE TO NEXT PAGE FOR INTRODUCTION TO THE LAST TWO PROBLEMS

The last two problems involve an algorithm that I will call `SimplexNBF`. As we did with `SimplexBF`, you will first try this algorithm "by hand" (using Python to compute pivots) and then we'll come back and see why it works. `SimplexNBF` starts with a Canonical Max tableau that is *not* basic-feasible (that's the "N") and does pivots (variable-swaps) to create an equivalent tableau that *is* basic-feasible. Here's a description of `SimplexNBF` along with an example.

Our initial tableau is not basic feasible, which means there is at least one $b_i < 0$. I'll demonstrate the algorithm for this Canonical Max tableau:

x_1	x_2	-1	
$-1/4$	-1	-2	$= -t_1$
-2	-1	-6	$-t_2$
-1	-1	-4	$= -t_3$
3	4	48	$= -t_4$
-1	-2	0	$= obj$

Step 1 (`SimplexNBF` version of "Target"): choose the $b_i < 0$ that is furthest down the last column. Let's say that occur in row "q".

For the tableau above, the Target would be $b_3 = -4$, so $q = 3$.

Step 2 (`SimplexNBF` version of "Candidates"): for the row q chosen in the Target step, choose any $a_{qj} < 0$ in this row; this becomes a candidate. In addition, any a_{rj} that is in the same column as a_{qj}, *and* below a_{qj}, *and* has $a_{rj} > 0$, is also a candidate.

For the tableau above, recall that we had a Target in row 3. We could choose $a_{32} = -1$ as a candidate since it is in row 3 and satisfies $a_{32} < 0$. If we make this choice, then note that below a_{32} in column 2, we also have $a_{42} = 4$, which is > 0, so that becomes a candidate as well.

Step 3 (`SimplexNBF` version of "Select"): scanning over all the Candidates a_{ij} from Step 2, compute the ratios b_i/a_{ij} (that b_i is the "b" in the same row as a_{ij}). Identify the minimum of these ratios, i.e., find a_{kj} from among the Candidates such that b_k/a_{kj} is $\leq b_i/a_{ij}$ for all other Candidates a_{ij}.

For the tableau above, we identified the Candidates as a_{32} and a_{42}, so we compute $b_3/a_{32} = (-4)/(-1) = 4$ and $b_4/a_{42} = 48/4$. The smaller of these is b_3/a_{32}, so we select a_{32}.

Step 4 ("Pivot": pivot the tableau on the a_{kj} that you Selected in Step 3.

In the new tableau computed in Step 4, check if all the b_j are ≥ 0.

Once you complete Step 4,

If all $b_j \geq 0$, the tableau is now basic-feasible, so switch to `SimplexBF` to finish the problem.

If some $b_j < 0$, go back and repeat Steps 1 through 4, repeatedly, until all the b_j are ≥ 0.

CONTINUE TO NEXT PAGE FOR THE LAST TWO PROBLEMS IN WHICH YOU WILL WORK THROUGH THIS ALGORITHM. USE MY PIVOT CODE TO DO ALL THE PIVOTING.

5 (COMPUTATION BUT NOT CODING): Download the pivot notebook from Moodle (Python section), upload it to either the Jupyterhub or `colab.research.google.com`, and edit the last cell to solve the Canonical Max problem described by the tableau:

x	y	-1	
-1	-1	-2	$= -t_1$
1	-2	0	$= -t_2$
-2	1	1	$= -t_3$
-3	1	0	$= obj$

You will need to call `pivot` some number of times, where you choose the entries to pivot on based on the rules of `SimplexNBF` and/or `SimplexBF`. At the end, insert a text cell that indicates what the maximum value is and at what (x, y) it occurs.

6 (COMPUTATION BUT NOT CODING): Download the pivot notebook from Moodle (Python section), upload it to either the Jupyterhub or `colab.research.google.com`, and edit the last cell to solve the following problem.

You run a farm that each week produces 30 lbs of tomatoes, 20 lbs of peppers, and 40 lbs of kale. You can use these ingredients to produce three different products; the table below shows each product, the ingredients it requires, the time required to make it, and the profit it generates when you sell it to Hometown Market, the store down the street from your farm:

Product	Tomatoes used	Peppers used	Kale used	Time needed to make	Profit
Pizza (1)	0.5 lbs	0.3 lbs	0.1 lbs	1.5 hr	\$ 5
Soup (1 cup)	0.3 lbs	0.1 lbs	0.4 lbs	0.1 hr	\$ 1
Stuffed Pepper (1)	0.1 lbs	0.8 lbs	0.05 lbs	0.25 hr	\$ 1.50

A few other facts about your situation:

- You only have a total of 40 hrs each week to make these products.
- You have promised Hometown Market that each week you will provide at least 30 "units" of pizzas and stuffed peppers in whatever combination you choose. So, if x_1 is the number of pizzas you produce each week, and x_3 the number of stuffed peppers you produce each week, you must have $x_1 + x_3 \geq 30$.
- At the end of the week, you can sell any tomatoes that you have left over for \$ 2/lb at the local farmers' market. (You can not sell any leftover peppers or kale.)

How many pizzas, cups of soup, and stuffed peppers should you produce in order to maximize your profit (including the money you earn from selling the leftover tomatoes)?

You will need to call `pivot` some number of times, where you choose the entries to pivot on based on the rules of `SimplexNBF` and/or `SimplexBF`. At the end, insert a text cell that indicates what the maximum profit is and how many pizzas, cups of soup, and stuffed peppers you produce to achieve that profit.