CSC 284/484 Advanced Algorithms - homework 2
applied homework due: February 20th, 11:59pm EST
theoretical homework due: February 20th, 12:30pm EST

Do not search for a solution online, do not use any written material when writing any part of the code (for example, no copy-paste, no open textbook when writing code, no reediting of an old source file from an old project, etc). You can discuss your solution(s) or problems/issues you are facing with your classmates. Do not share code.

PROBLEM 4 (applied)—implement the first phase of the simplex algorithm

- Test your algorithm on the attached test data.
- Write a little note indicating:
  - Whether your code passed all the tests (if not where is the problem).
  - Timing information (how long your code runs on each test).
- Your note should be typeset (LaTeX, Word, etc); send your code and note to the instructor and the TA (`breber@cs.rochester.edu`) (subject line: CSC 284/484 HOMEWORK 1).
- Allowed languages: C, C++, Java, Python (I discourage you from using Python—it is better for you to learn how to code algorithms in more efficient languages).

The objective is to find a feasible solution of the following "feasibility" linear program ("feasibility" linear program is one without an objective function—we just want a feasible solution of the program (feasible = one that satisfies all the constraints)):

$$Ax = b, x \geq 0.$$

You can assume that $b \geq 0$ (somebody already multiplied all the equalities by $+1$ or -1 so that $b \geq 0$).

INPUT FORMAT: The first line contains an integer k—the number of problem instances. Then the descriptions of the k problem instances follow. For each instance the first line contains two numbers n, m, where n is the number of variables and m is the number of constraints. Then m lines follow, each line containing $n+1$ numbers—each line gives a row in A and the corresponding $b_i \geq 0$. We have $1 \leq n \leq 1,000$ and $1 \leq m \leq 1,000$.

OUTPUT FORMAT:
The output contains one or two lines for each problem.

- If the problem is infeasible then there is a line saying INFEASIBLE.
- If the problem is feasible then the line contains a feasible basis; the indices of the basis are in $\{1, \ldots, n\}$, listed in increasing order.

EXAMPLE INPUT:

```
3
3 2
1 1 0 2
1 0 -1 3
2 2
1 1 2
2 2 3
4 3
5 -8 -7 6 0
0 -6 3 -7 10
-2 -8 2 9 8
```

EXAMPLE OUTPUT:

```
INFEASIBLE
INFEASIBLE
1 3 4
```

PROBLEM 5 (theoretical)—solve a problem using LP

Typeset your solution (LaTeX, Word, etc) and send pdf to the instructor by email (subject line: CSC 284/484 HOMEWORK 1). In addition to sending the pdf print your solution and bring it to class of Feb. 20.

Pat and Mat are driving around the country. Every time they **stop** in a city they write down the value shown by the odometer. The odometer shows the distance traveled (in parasangs) since the car was made rounded down to the closest integer. Looking at their travel log they would like to infer an upper bound on the distance between any two cities. The distance is not necessarily symmetric—the distance from a to b can be different from b to a. Pat and Mat can go through intermediate cities between their consecutive stops (without stopping in the intermediate cities) and they always take the shortest path between the consecutive stops.

You do *not* have to implement the problem/solution. The description below is just to further clarify the input (and give you examples).

INPUT FORMAT: The first line contains the number of test cases. Each test case is described on several lines. The first line contains n (the number of cities) and m (the number of visits on their journey). The second line contains m numbers from $\{1, \ldots, n\}$—the cities visited (the cities may repeat). The third line contains m integers—the readings of the odometer on their visits. You can assume $2 \leq n \leq 10$ and $2 \leq m \leq 400$.

OUTPUT FORMAT: The output is an $n \times n$ matrix (with zero diagonal) output in n rows with n numbers per row. The i, j-th entry of the matrix is the best possible upper bound on the distance between city i and city j that can be inferred from the travel log. (The numbers are output as follows. If the number is an integer then the integer is output. If the number is not an integer then it is output in the form p/q or $-p/q$ where p, q are integers such that $\gcd(p, q) = 1$.)

EXAMPLE INPUT:

```
1
4 10
1 3 4 1 3 4 1 2 1 2 1
0 3 5 6 9 11 12 15 19 22 26
```

EXAMPLE OUTPUT:

```
0 4 4 6
5 0 17/2 21/2
4 7 0 3
2 5 5 0
```