02-713 Homework #9: Network Flow

Due: Apr. 16 by 9:30am

Don't use last year's solutions. Don't look up solutions in the web. The point of these homeworks is to prepare for the exams.

You may talk with your classmates about these problems, but you **must write up your solutions independently**, without using common notes or worksheets. You must indicate at the top of your homework with whom you worked.

Your write up should be clear and concise. It should be submitted via Autolab (`https://autolab.cs.cmu.edu/02713-s14/`) as a typeset PDF.

1. You are designing a experiment in which you want to measure certain properties $p_1, \dots, p_n$ of a yeast culture. You have a set of tools $t_1, \dots, t_m$ that can each measure a subset S_i of the properties. For example, tool t_i measures S_i which may equal $\{p_7, p_8\}$. To be sure that your results are not due to noise or other artifact, you must measure every property at least k times using k different tools.

   (a) Give a polynomial-time algorithm that decides whether the tools you have are sufficient to measure the desired properties the desired number of times.

   (b) Suppose now each tool t_i comes from manufacturer M_i and we have the additional constraint that the tools to test any property p_i can't all come from the same manufacturer. Give a polynomial-time algorithm to solve this problem.

2. Consider the Binary Magic Square problem: Given a list of n integers $\vec{r} = (r_1, \dots, r_n)$ and a list of m integers $\vec{c} = (c_1, \dots, c_m)$, we ask whether there is an $n \times m$ grid of 0's and 1's such that row i sums to r_i and column j sums to c_j. We assume $\sum_i r_i = \sum_j c_j$.

   Examples: $n = m = 3$ with $\vec{c} = (1, 2, 0)$ and $\vec{r} = (1, 1, 1)$ (answer=yes) or $\vec{r} = (3, 0, 0)$ (answer=no):

   Yes

1	0	0	1
0	1	0	1
0	1	0	1
1	2	0	

   No

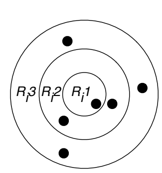
1	1	1	3
			0
			0
1	2	0	

   Use Network Flow to create a polynomial-time algorithm to decide whether it is possible to design a 0/1 grid that obeys the given $\vec{r}$ and $\vec{c}$ sums.

3. You are deploying n cheap temperature-measurement devices in the field, with device t_i at coordinates (x_i, y_i), measured in meters from some arbitrary point. These devices record their temperature over several weeks. The devices are likely to fail so you want to design a system to back up the data they have collected in the following way: Each device has a radio transmitter that can reach d meters. When a device t_i senses it is about to fail, it will transmit its data, and that data should reach at least k other devices. Each device can serve as the backup for at most b other devices.

   (a) Design a polynomial-time algorithm to determine whether the given positions of the devices meets the requirements and, if it does, to output the set B_i of k back up devices for every device.

   (b) Suppose for every device t_i, we are now given a collection of sets R_i^d for $d = 1, \dots, k$, where R_i^d contains the set of devices that are at distance ring d from t_i. See figure below:



We add the following requirement: each of the devices in the backup set B_i for device t_i must come from a different ring R_i^d. (That is, we need a very close device from ring R_i^1 and a slightly farther device from R_i^2, etc.) Give a polynomial time algorithm to find the backup sets that meet this requirement.