CS325: Analysis of Algorithms, Fall 2016

Group Assignment 4

Due: Thr, 12/1/16

Homework Policy:

1. Students should work on group assignments in groups of preferably three people. Each group submits to TEACH a zip file that includes their source code and their report. Each group, also, hands in a printed hard copy of the report in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded, and the codes submitted to teach will be tested.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. You can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. More items might be added to this list. ☺

In this assignment we work on the following two problems.

2SAT. We have seen in class that 3SAT is NP-hard, but 1SAT can be solved in polynomial time. In fact, there is a polynomial time algorithm for 2SAT, too. Given a 2CNF formula of length n, design an $O(n)$ time algorithm to decide if it is satisfiable or not. (Hint: let x be any variable in the formula, set x to be true, check if it implies a contradiction or not.)

Inside/outside. There is a town with n houses: $\{0, \ldots, n-1\}$ that are located on a ring road in this order. People of the town decided to build m new direct roads between pairs of houses. They came up with a list of pairs $\{(p_1, q_1), \ldots, (p_m, q_m)\}$, where p_i's and q_i's are house numbers (from $\{0, \ldots, n-1\}$). The only problem they have is that they do not know how to build a junction or a bridge. So, they can only accomplish the task if it is possible to build these roads with no crossings. For each new road they have two options: to build it inside the ring road, or outside the ring road. Your job is to decide if it is possible to build the new roads, and you really want to use your $2SAT$ solver. See Figure 1 for an example.

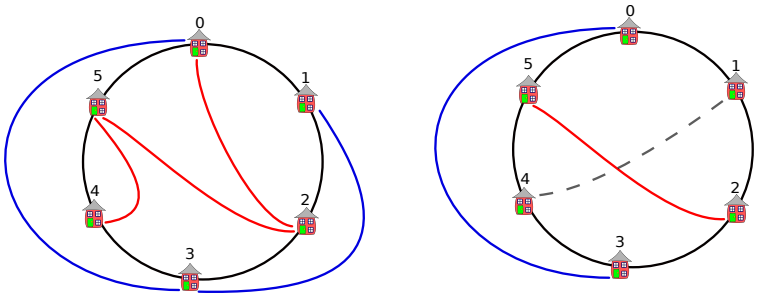

Figure 1: Left: The roads for the case that $n = 6$, $m = 5$, and the pairs are $\{(0,3),(1,3),(0,2),(2,5),(4,5)\}$. Right: it is not possible to build roads for the case that $n = 6$, $m = 3$, and the pairs are $\{(0,3),(2,5),(1,4)\}$.

Report

(1) Describe a $O(n)$ time algorithm for the 2SAT problem. Analyze the running time of your algorithm to show it is $O(n)$.

(2) Describe a reduction from the "Inside/outside" problem to 2SAT. What is the running time of your reduction? Show that your reduction is correct: the 2CNF instance that you build is satisfiable if and only if there is a way to build noncrossing roads.

Code Write a program to solve the "inside/outside" problem. The input to your program consists of two lines. The first line contains two numbers (separated by comma) $0 \leq n \leq 1000$ and $0 \leq m \leq 1000$. The second contains $2m$ numbers (separated by commas), $p_1, q_1, p_2, q_2, \ldots, p_m, q_m$ in this order. All p_i's and q_i's are from $\{0, \ldots, n-1\}$.

The output file contains exactly one word, it is "YES" if it is possible to connect the pairs of houses, and "NO", otherwise.

Sample Input (1):
6,5
0,3,1,3,0,2,2,5,4,5

Sample Output (1):
YES

Sample Input (2):
6,5
0,3,2,5,1,4

Sample Output (1):
NO

Submission Each group submits to TEACH a zip file that includes their source code (*which must be just one file with name "ring.cpp", "ring.java", or "ring.py"*) and their report. This file can be submitted by any member of the group, but all names must be listed in the submitted report. Each group, also, hands in a printed hard copy of the report in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded, and the codes submitted to teach will be tested.

Test your code with the sample test files (`http://web.engr.oregonstate.edu/~nayyeria/CS325/Fall16/hws/test4.zip`) before submitting them, to make sure there is no formatting error.

Do *not* submit a solution for the following problem, it is for practice.

Practice Problem. [1]

(a) Describe a polynomial-time reduction from Partition to SubsetSum.

(b) Describe a polynomial-time reduction from SubsetSum to Partition.

[1] This problem is from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on NP-hardness is recommended.