CS 336: Algorithms **Problem Set 3**
Date: Saturday, October 9, 2021 **Due:** Thursday, October 14, 2021

Submit your solution on Canvas.

Problem 1. You are asked to design a computer program for an online advertising start up. This start up connects website owners with advertisers. When a user visits a web page, your program should retrieve a list of relevant ads from a database and decide which ads to display. Your goal is to maximize the expected revenue received from the ads.

The program is given a list of n ads; and a list of m ad slots on the web page. The slots are located at different places on the web page, and the user may see some slots and not others. For every slot i, you are given a probability a_i that the user will see that slot. For every ad j, you are given the following information:

- the revenue r_j that the web owner receives from the advertiser if the user clicks on the ad and
- the probability b_j that the user clicks on the ad if he or she sees the ad.

If ad j is displayed in slot i, the expected revenue from this ad equals $a_i b_j r_j$. The program needs to assign an ad to every available slot. Every ad may be displayed at most once.

Formally, you need to find an *injective* mapping f from the set of slots to ads (f is *injective* if $f(i') \neq f(i'')$ for distinct i' and i'') so as to maximize the following sum:

$$\sum_{i=1}^{m} a_i b_{f(i)} r_{f(i)}.$$

You can assume that $n \geq m$. Hint: First, assume that $n = m$.

I. What is a feasible solution for this problem?

II. Design and describe a **greedy** algorithm for this problem.

III. Analyze its running time. To get a full credit for the problem, the running time of the algorithm must be at most $O(n \log n + m \log m)$.

IV. Prove that the algorithm is correct.

Problem 2. A mobile communications company, Tech336 wants to upgrade their cellular network along highway X to a new wireless standard 10G. To do so, it plans to install new wireless transmitters on some of their existing towers. A user can receive a signal from a tower if the distance from the user to the tower is at most `range`. To guarantee an uninterrupted service for all their customers, Tech336 wants to install new transmitters so that no matter where the customer is located on the highway, he or she is within the range of at least two transmitters. Your goal is to find the minimum number of transmitters required for the upgrade.

Design and implement a greedy algorithm for finding the minimum number of transmitters. Note that Tech336 can install at most one transmitter on an existing a cell tower. Write the following function

```
MinStations(int roadLength, int range, std::vector<int> towers)
```

The parameters of this function are `roadLength` – the length of the highway; `range` –the transmission range of the new transmitters; `towers` – positions of the towers. Each position is the distance from the tower to the beginning of the highway. Function `MinStations` should return the minimum number of new transmitters. If the problem has no feasible solution, `MinStations` should return -1. In this problem, you can assume that the highway is a straight line.

Collaboration policy for Problems 1 and 2: Please, solve these problems on your own. Do not collaborate with other students.

Instructions for the programming assignment.

Download files

- `student_code_3.h` – this file should contain your solution.
- `problem_solver_3.cpp` – this is the main file in the project (don't edit this file!).
- `test_framework.h` – this is a library responsible for reading and writing data files (don't edit this file!)
- `small_problem_set_3.in` and `large_problem_set_3.in` – these files contain test problems for your algorithm (don't edit these files!)

Place all files in a new folder/directory. Write your code in function `MinStations`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_3.h`. Compile and run your code. To compile your code do the following.

- If you use Clang compiler, type
  `clang++ -std=c++17 -pedantic-errors problem_solver_3.cpp -O2 -o problem_solver_3`
- If you use GNU C++ compiler, type
  `g++ -std=c++17 -pedantic-errors problem_solver_3.cpp -O2 -o problem_solver_3`
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  `cl /EHsc problem_solver_3.cpp`

Your compiler should be compatible with `C++17`. If you work in the Wilkinson Lab, you need to start developer tools first: Type

- `scl enable devtoolset-4 bash`

Once you compile your code, start your program. Type `./problem_solver_3 small` to run your code on simple problems and `./problem_solver_3 large` to run your code on hard problems. On Windows, type `problem_solver_3.exe small` and `problem_solver_3.exe large`, respectively. Make sure that executable is located in the same folder as files `small_problem_set_3.in` and `large_problem_set_3.in`. If your code works correctly, you will get the following message:

> Problem set 3. Your algorithm solved all test problems correctly. Congratulations! `solution_3.dat` via Canvas.

If your code makes a mistake, you may get a message like this:

> `Problem set 3. Mistake in problem #15. Correct answer: 4. Your answer: 12.`

Please, test your code with the both problem sets (small and large). When your code is ready, submit file `student_code_3.h` on Canvas. Make sure that you are submitting the latest versions!