

Submit your solution on Canvas.

Do not discuss these problems with other students. You should solve these problems on your own.

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 $n \geq m$.

I. Design and describe a greedy algorithm for this problem.

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

III. Prove that the algorithm is correct.

Problem 2. In this exercise, you need to implement the greedy algorithm from Problem 1.

- double FindBestAssignment (std::vector<double> a , std::vector<double> b ,
std::vector<double> r)

This function should return the maximum expected revenue for a given page i.e.,

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

where f is an injective mapping.

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!)
- `problem_set_3.in` – this file contains test problems for your algorithm (don't edit this file!)

Place all files in a new folder/directory. Write your code in the function `FindBestAssignment`. 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 GNU C++ compiler, type
`g++ -std=c++11 problem_solver_3.cpp -o problem_solver_3`
- If you use CLang compiler, type
`clang++ -std=c++11 problem_solver_3.cpp -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++11. If you work in TLab, 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` on Unix or Mac and `problem_solver_3.exe` on Windows. Make sure that the executable is located in the same folder as file `problem_set_3.in`. Your program will generate `solution_3.dat` that contains solutions to the problems from file `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!
- Don't forget to submit your source code and file `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.

Finally, when your code is ready, submit files `student_code_3.h` and `solution_3.dat` via Canvas. Make sure that you are submitting the latest versions.

Remark: If you want to debug your code, please, type `./problem_solver_3 15` on Unix or Mac and `problem_solver_3.exe 15` on Windows. This command will call your function only on one problem – the problem #15 and thus let you debug your code on the problem where your program erred. Note that this command will not generate or update `solution_3.dat`. So before submitting your solution, you need to run your program without any command line arguments.