

**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 given a set of red and blue points on a real line:  $red[0], \dots, red[n-1]$  and  $blue[0], \dots, blue[n-1]$ . Your goal is to find a minimum cost perfect matching between them i.e., a *one-to-one* function  $f: \{0, \dots, n-1\} \rightarrow \{0, \dots, n-1\}$  that minimizes the following expression:

$$\text{cost}(f) = \sum_{i=0}^{n-1} |red[i] - blue[f(i)]|. \quad (1)$$

- I. Design and describe an algorithm that solves this problem.
- II. Prove that it is correct.
- III. Analyze its running time.

**Problem 2.** In this assignment, we ask you to implement your algorithm from the previous problem. Write the following function:

- `int MinMatching(std::vector<int> blue, std::vector<int> red)`

Array *red* contains coordinates of red points; array *blue* contains coordinate of blue points. The function should return the cost of the matching defined as (1).

**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 function `MinMatching`. 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 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` 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.