Algorithms EECS 336 **Problem Set #4**
Date: April 29, 2018 **Due:** May 13, 2018

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

Problem 1 (Quiz)**.** Please, submit your answers to the questions below under the Quizzes section of Canvas.

I. In this exercise, we ask you to construct a Huffman code[1] for letters "A", "B", "C", "D", and "E". The letter frequencies are given in the following table.

Letter	A	B	C	D	E
Frequency	0.05	0.25	0.45	0.15	0.1

1. How many characters do you need to represent the word "ABCDE" using this code?

2. How many characters do you need to represent the word "CE" using this code?

II. Construct a Huffman code[1] for letters "N", "P", "Q", "R", and "U". The letter frequencies are given in the following table.

Letter	N	P	Q	R	U
Frequency	0.1	0.3	0.2	0.1	0.3

3. How many characters do you need to represent the word "NU" using this code?

4. How many characters and ones do you need to represent the word "PQR" using this code?

III. Find the minimum weighted completion time for Job Sets #1 and #2 (see figures on the last page).

5. Minimum weighted completion time for Job Set #1.

6. Minimum weighted completion time for Job Set #2.

Problem 2. You are given a set of red and blue points on a real line: $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. Your goal is to find a minimum cost perfect matching between them i.e., a one-to-one function $f : \{1, \ldots, n\} \to \{1, \ldots, n\}$ that minimizes the following expression:

$$\text{cost}(f) = \sum_{i=1}^{n} \left| x_i - y_{f(i)} \right|. \tag{1}$$

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

Problem 3. In this assignment, we ask you to implement your algorithm from the previous problem. Write the following function:

- `int MinMatching (std::vector<int> x, std::vector<int> y)`

[1] We will discuss Huffman coding on Tuesday, May 1

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

Instructions for the programming assignment. Download files:

- `student_code_4.h` – this file should contain your solution.
- `problem_solver_4.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_4.dt` – 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 `MinMatching`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_4.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_4.cpp -o problem_solver_4
```
- If you use CLang compiler, type
```
clang++ -std=c++11 problem_solver_4.cpp -o problem_solver_4
```
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
```
cl /EHsc problem_solver_4.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_4` on Unix or Mac and `problem_solver_4.exe` on Windows. Make sure that the executable is located in the same folder as file `problem_set_4.dt`. Your program will generate `solution_4.dat` that contains solutions to the problems from file `problem_set_4.dt`. If your code works correctly, you will get the following message:

- Problem set 4. Your algorithm solved all test problems correctly. Congratulations!
- Don't forget to submit your source code and file `solution_4.dat` via Canvas.

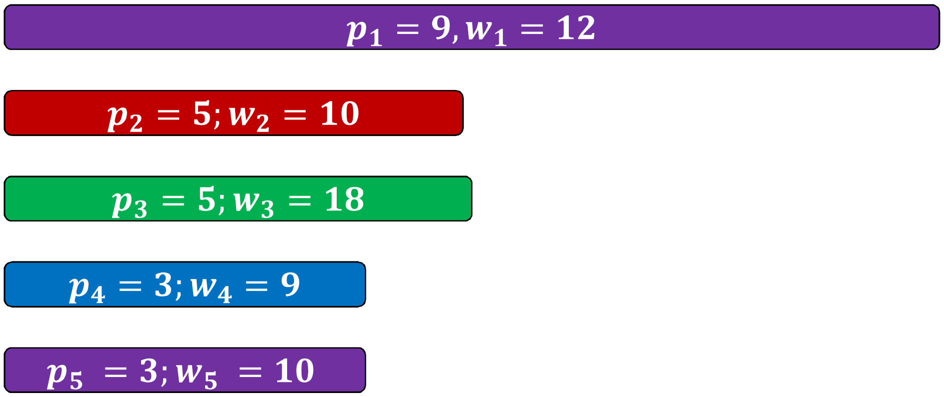
If your code makes a mistake, you may get a message like this:

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

Finally, when your code is ready, submit files `student_code_4.h` and `solution_4.dat` via Canvas. Make sure that you are submitting the latest versions.

Remark: If you want to debug your code, please, type `./problem_solver_4 15` on Unix or Mac and `problem_solver_4.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_4.dat`. So before submitting your solution, you need to run your program without any command line arguments.

# Job Set #1
for Sum of Weighted Completion Times



$p_1 = 9, w_1 = 12$

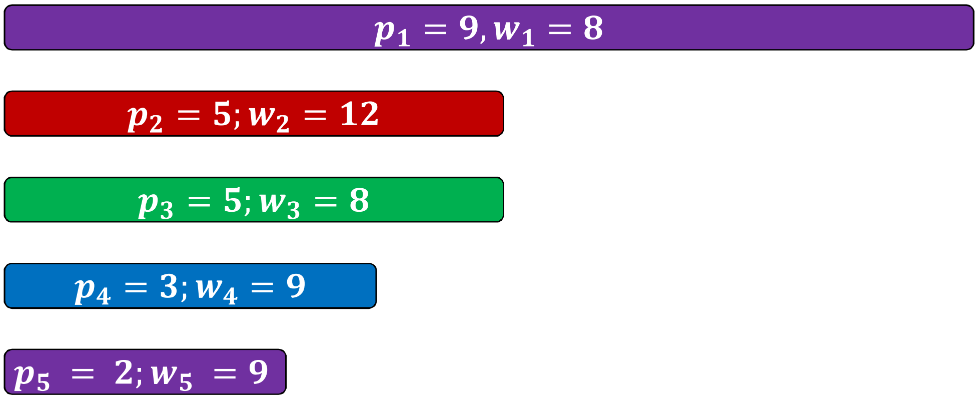
$p_2 = 5; w_2 = 10$

$p_3 = 5; w_3 = 18$

$p_4 = 3; w_4 = 9$

$p_5 = 3; w_5 = 10$

# Job Set #2
for Sum of Weighted Completion Times



$p_1 = 9, w_1 = 8$

$p_2 = 5; w_2 = 12$

$p_3 = 5; w_3 = 8$

$p_4 = 3; w_4 = 9$

$p_5 = 2; w_5 = 9$