Algorithms EECS 336 **Problem Set #4**
Date: Wednesday, January 30, 2019 **Due:** Tuesday, February 5 (subject to change), 2018

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

Problem 1. In this exercise, we consider two variants of the following problem. You are given an array of integer numbers $X[0], \dots, X[n-1]$. You need to partition it into k groups $P_1, \dots, P_k$ so that numbers in every group P_j are distinct. Your goal is to minimize the number of groups, k.

> **Problem 1A:** In the first variant of the problem, each group P_j must contain consecutive elements of array X. For example, we can partition array $1, 2, 3, 4, 1, 7$ into groups $G_1 = \langle 1, 2\rangle$ and $G_2 = \langle 3, 4, 1, 7\rangle$ or $G_1 = \langle 1, 2, 3\rangle$ and $G_2 = \langle 4, 1, 7\rangle$, but we cannot partition it into groups $G_1 = \langle 1, 3, 7\rangle$ and $G_2 = \langle 2, 4, 1\rangle$, because elements in these groups are not consecutive. We also cannot partition X into $G_1 = \langle 1, 2, 3, 4, 1\rangle$ and $G_2 = \langle 7\rangle$, because in this case G_1 contains two identical elements (specifically, two "1"s).
>
> **Problem 1B:** In the second variant of the problem, groups G_i do not have to contain consecutive elements of array X. So in the above example, the first three partitionings $G_1 = \langle 1, 2\rangle$ and $G_2 = \langle 3, 4, 1, 7\rangle$; $G_1 = \langle 1, 2, 3\rangle$ and $G_2 = \langle 4, 1, 7\rangle$; $G_1 = \langle 1, 3, 7\rangle$ and $G_2 = \langle 2, 4, 1\rangle$ are valid. However, the fourth partitioning $G_1 = \langle 1, 2, 3, 4, 1\rangle$ and $G_2 = \langle 7\rangle$ is still not valid.

For both problems – Problem 1A and Problem 1B – do the following:

1. Describe a greedy algorithm that solves this problem.
2. Prove that your algorithm is correct.
3. Analyze the running time of your algorithm. To get a full credit for this problem, the running time your algorithm should be $O(n \log n)$ or less.

Problem 2. We ask you to implement your algorithms for Problem 1A and Problem 1B:

- `int ProblemA (std::vector<int> X)`
- `int ProblemB (std::vector<int> X)`

Array X is the array you need to partition. `ProblemA` and `ProblemB` should return the number of groups in the optimal solutions to Problem 1A and Problem 1B.

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.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 functions `ProblemA` and `ProblemB`. 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.in`. Your program will generate `solution_4.dat` that contains solutions to the problems from file `problem_set_4.in`. 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.