Algorithms EECS 336 **Problem Set #6**
Date: May 21, 2018 **Due:** May 28, 2018

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

Problem 1. This problem consists of two parts.

I. We say that a sequence $a_1, \ldots, a_n$ is balanced if $\sum_{i=1}^{n} a_i = 0$. Your goal is to design a dynamic programming algorithm that given a sequence of integer numbers $x_1, \ldots, x_n$ finds a balanced sequence $a_1, \ldots, a_n$ in which each a_i is -1, 0, or 1 that maximizes

$$\sum_{i=1}^{n} a_i x_i.$$

1. Describe a subproblem for this problem.
2. Write a recurrence formula (or algorithm) for computing the value of this subproblem and explain why this formula is correct.
3. Write a DP algorithm for the problem. The running time of the algorithm must be polynomial in n but you do not need to optimize the running time.

II. Design a dynamic programming algorithm that solves the previous problem for L-balanced sequences. We say that a sequence $a_1, \ldots, a_n$ is L-balanced if for all t_1 and t_2 ($t_1 \leq t_2$) we have

$$\sum_{i=t_1}^{t_2} a_i \leq L.$$

The algorithm receives $x_1, \ldots, x_n$ and L. Note that you cannot change the order of x_i.

1. Describe a subproblem for this problem.
2. Write a recurrence formula (or algorithm) for computing the value of this subproblem and explain why this formula is correct.
3. Write a DP algorithm for the problem. The running time of the algorithm must be polynomial in n but you do not need to optimize the running time.

Problem 2. Implement your algorithm from Problem 1 part I.

- `int FindMaxBalancedSequence (const std::vector<int>& weights)`

Array `weights` contains the set of weights x_i.

Instructions for the programming assignment. Download files:

- `student_code_6.h` – this file should contain your solution.
- `problem_solver_6.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_6.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 `FindMaxBalancedSequence`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_6.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_6.cpp -o problem_solver_6`
- If you use CLang compiler, type
  `clang++ -std=c++11 problem_solver_6.cpp -o problem_solver_6`
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  `cl /EHsc problem_solver_6.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_6` on Unix or Mac and `problem_solver_6.exe` on Windows. Make sure that the executable is located in the same folder as file `problem_set_6.dt`. Your program will generate `solution_6.dat` that contains solutions to the problems from file `problem_set_6.dt`. If your code works correctly, you will get the following message:

- Problem set 6. Your algorithm solved all test problems correctly. Congratulations!
- Don't forget to submit your source code and file `solution_6.dat` via Canvas.

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

- `Problem set 6. Mistake in problem #15. Correct answer: 4. Your answer: 12.`

Finally, when your code is ready, submit files `student_code_6.h` and `solution_6.dat` via Canvas. Make sure that you are submitting the latest versions.

Remark: If you want to debug your code, please, type `./problem_solver_6 15` on Unix or Mac and `problem_solver_6.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_6.dat`. So before submitting your solution, you need to run your program without any command line arguments.