Algorithms EECS 336 **Problem Set #1**
Date: January 8, 2018 **Due:** 11:59pm, January 14, 2018

Submit your solution on Canvas.

Definition 1. *Suppose f and g are positive functions from $\mathbb{N}$ to $\mathbb{N}$. We say that $f(n) = O(g(n))$ if there exists a positive constant C such that for every n in $\mathbb{N}$, we have $f(n) \leq Cg(n)$.*

Problem 1 (Quiz)**.** I. Answer the following questions. Enter your answers under the Quizzes section of Canvas. We will open the Canvas submission page on January 10.

1. True or False: $n = O(n \log_2 n)$.
2. True or False: $n^2 = O(n)$.
3. True or False: $n \log_2^5 n = O(n^2)$.
4. True or False: $\log_2 n = O(\log_e n)$.
5. True or False: $\log_e n = O(\log_2 n)$.
6. True or False: $2^n = O(n^{\log_2 n})$.
7. True or False: $n^{\log_2 n} = O(2^n)$.
8. True or False: For all positive functions f and g, if $f(n) = O(g(n))$, then $n \cdot f(n) = O(n \cdot g(n))$.
9. True or False: For all positive functions f and g, if $f(n) = O(g(n))$, then $f(n)+n = O(g(n)+n)$.

Problem 2 (**Warmup programming exercise. You can discuss this problem with your classmates. However, you must acknowledge any collaboration.**)**.** A group of students at Northwestern University (NU) want to find out how popular Northwestern is. In order to do so, they plan to write a program that reads news on the Internet and counts the number of occurrences of the word "NU" (the letters must be capitalized). Your goal is to write a function with the following signature/declaration:

- `int CountNUOccurrences(const std::string& message)`

that counts and returns the number of substrings "NU" in the string `message`.

Instructions for the programming assignment. Download files:

- `student_code_1.h` – this file should contain your solution.
- `problem_solver_1.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_1.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 `CountNUOccurrences`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_1.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_1.cpp -o problem_solver_1`
- If you use CLang compiler, type
  `clang++ -std=c++11 problem_solver_1.cpp -o problem_solver_1`
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  `cl /EHsc problem_solver_1.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_1` on Unix or Mac and `problem_solver_1.exe` on Windows. Make sure that the executable is located in the same folder as file `problem_set_1.in`. Your program will generate `solution_1.dat` that contains solutions to the problems from file `problem_set_1.in`. If your code works correctly, you will get the following message:

- Problem set 1. Your algorithm solved all test problems correctly. Congratulations!
- Don't forget to submit your source code and file `solution_1.dat` via Canvas.

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

- `Problem set 1. Mistake in problem #15. Correct answer: 4. Your answer: 12.`

Finally, when your code is ready, submit files `student_code_1.h` and `solution_1.dat` via Canvas. Make sure that you are submitting the latest versions.

Remark: If you want to debug your code, please, type `./problem_solver_1 15` on Unix or Mac and `problem_solver_1.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_1.dat`. So before submitting your solution, you need to run your program without any command line arguments.