CS507 Computing Foundation for Computational Science HW6

Min Long

This HW is a pure written HW, and should be finished in a PDF file. To make a PDF file, you can scan your solutions, or type your solution in WORD and then convert it.

Due time: Due: 11/30/2022, 23:59
Total: 40 points
Submission command: `submit minlong CS507 HW6`

Instruction to electronic submission: http://cs.boisestate.edu/~cs221/SubmissionProcedure.html

- The written assignment can be done in a pure text format (*.txt, for example, problem1.txt) on Onyx.
- The programming assignment should be presented with source codes.
- Each problem should have its own working directory, such as HW2/prob1, HW2/prob2 ... For example, the following table shows the structure of HW1 from the user "student1" and how to submit HW to us through Onyx

```
[student1@onyx:HW1]$ ls -l
drwxr-x---. 2 student1 Students  13 Sep 19  2021 prob1
drwxr-xr-x. 3 student1 Students  14 Sep 19  2021 prob2
[student1@onyx:HW1]$ cd prob1/
[minlong@onyx:prob1]$ ls
-rw-r-----. 1 student1 Students 943 Sep 19  2021 problem1.txt
$ cd ..
[student1@onyx:HW1]$ pwd
/home/student1/CS507/HW1
[student1@onyx:HW1]$ submit minlong CS507 HW1
```

Listing 1: A sample structure of homework and submission procedure.

- Your source codes (if any) must compile and run on Onyx.
- Documentation is important and proper comments are expected in your source code.
  - comments giving description of: purpose, parameters, and return value if applicable
  - other comments where clarification of source code is needed
  - proper and consistent indentation
  - proper structure and modularity

Don't ask us or your classmates directly for solutions (it happened); just try as much as possible. Be patient and enjoy coding!

Written Problems

1. (10 pts = 2pts ×5) **Informal Definitions of Asymptotic Notations**. Determine whether the following assertions are **true** or **false** using the **Limit Formula definitions** of the O, Ω, and Θ notations. Show all of your work.

   (1) $\frac{n(n+1)}{2} \in O(n^3)$

   (2) $\frac{n(n+1)}{2} \in \Theta(n^3)$

   (3) $\frac{n(n+1)}{2} \in \Theta(n^2)$

   (4) $\frac{n(n+1)}{2} \in O(n^2)$

   (5) $\frac{n(n+1)}{2} \in \Omega(n^2)$

2. (10 points): Running Time and Growth of Functions Assume evaluating a function $f(n)$ in the pseudocode below takes $\Theta(n)$ time, that is, $f(n) \in \Theta(n)$.

```
i = 1;
sum = 0;
while (i <= n)
       if (f(i) > k)
          then sum += f(i);
       i = 2*i;
```

What is the running time (use an asymptotic notation) of the above code? Justify your answer. **Hint:** You need to analyze the code line by line and count how much running time was spend on each of components (function call, loop...).

3. (10 pts) **Formal Definitions.** Prove or disprove the following assertions using the formal definitions of the notations involved. Show all of your work. Formal definition means we need to find c, n_0 to satisfy or dissatisfy the corresponding inequality.

   $4n^3 + 5n - 6 \in \Omega(n^2)$

4. (5 points) Explain why the statement, "The running time of an algorithm is $\Omega(1)$," is meaningless.

5. (5 pts) **Orders of Growth**. Order the following functions according to their orders of growth from lowest to highest.

   $5 \log(n + 100)^{10}$

   2^{2n}

   $0.001n^4 + 3n^3 + 1$

   $\log^2 n$

   $n^{1/3}$

   3^n