Homework 2
Com S 311
Due: Sep 16, 11:59PM

Late Submission Due: Sep 17, 11:59PM (25% penalty)

Outcomes.

- Determine whether a function is Big-O of other function or not
- Analyze asymptotic worst-case time complexity of algorithms

Note that submission instructions have changed. Please see guidelines.
There are 5 problems, each problem is worth 60 points.

1. Prove or disprove the following:

   (a) $5n^2 - 2n + 26 \in O(n^2)$.

   (b) $\forall a \geq 1 : a^n \in O(n!)$

   (c) $\forall a \geq 1 : 2^{n+a} \in O(2^n)$

   (d) $\forall a > 1 : (f(n) \in O(\texttt{log}_2 n)) \;\Rightarrow\; (f(n) \in O(\texttt{log}_a n))$

   (e) $2^n \in O(n^{\log^2 n})$.

   (f) $2^{2^{n+1}} \in O(2^{2^n})$.

2. (a) With respect to the input n, what is the worst-case time complexity of the following algorithm?

```
for (i = 1; i < exp(2, n); i = 2*i) {
    for (j = i; j > 1; j = ceil(j/2)) {
        <some-constant number of atomic/elementary operations>
    }
}
```

   In the above, `exp(2, n)` denotes 2^n and `ceil(j/2)` denotes $\lceil \frac{j}{2} \rceil$.

   (b) Consider the following method that computes the median of an array of consisting of distinct integers.

```
Input: Array a of size n. Assume that array has distinct elements.
```

```
for i  in [0,  n-1] {
    x = a[i];
    r = 0;
    for j in [0, n-1] {
        if (a[j] <x)
            r++;
    }
    if  r equals n/2 or (n+1)/2 return x;
 }
```

Determine the best-case and worst-case time complexities of this algorithm using big-oh notation. Provide a justification for your time-bounds. For a given array of size n, what is the best-case input? For a given array of size n, what is the worst-case input?

3. Given an array A containing 0s and 1s, such that all the 0s appear in the array before all the 1s. Write an algorithm with worst-case time complexity $O(\texttt{log}(n))$, which finds the smallest index i such that $A[i] = 1$. Describe your algorithm, and analyze its worst-case time complexity.

4. Assume that you are given an algorithm named `Merge` that can merge two sorted integer arrays of size n and m to generate a new sorted array of size $n + m$ in $O(m + n)$ time. Your task is to use `Merge` merge k sorted integer arrays each containing n elements, and output a single sorted array. Consider the following algorithm for this task. Let $A_1, A_2, \cdots, A_k$ be the input arrays.

```
A = empty array.
For i in range [1...k]
    A = Merge(A_i, A);

Return A;
```

Derive the worst-case asymptotic time complexity of the above algorithm.

5. Consider the following two methods that compute the greatest common divisor of two integers.

```
GCD(a, b) {

   n = min(a, b);

   for (int i = n; i >=1; i--)
      if  both a%i and b%i are zero, return i;

}
```

```
Fastgcd (a, b) {
   if b equals 0,
       return a;
   else
       return Fastgcd(b, a %b);
}
```

Write a Java program that implements both of the above methods. Play with the program by giving very large numbers as inputs to both the numbers.

(a) Pick two 9-digit primes and run both methods. Report the run time.

(b) Pick two 10 digit primes and run both methods. Report the run time.

(c) Show that the run time of of `FastGcd` is $O(n)$ when the inputs a and b are n-bit integers.

Please do not submit your code. You can use `System.currentTimeMillis()` to calculate run time of your methods.

You can look at `https://primes.utm.edu/lists/small/millions/` for 9 digit primes. You can look at `https://primes.utm.edu/lists/small/small.html` for 10 digit primes.

GUIDE LINES:

- Graded home works will be returned in the recitations. So please write your recitation number, time and TA name.
- It is important to know whether your really know! For each problem, if you write the statement "I do not know how to solve this problem" (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere between 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem.
- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.
- When proofs are required, you should make them both clear and rigorous. Do not hand waive.
- If you hand writing is not legible, then your homework will not be graded.
- Any concerns about grading should be made within one week of returning the homework.
- **Please submit your HW via blackboard. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solutions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. Name your file must be *YourNetID-HW2.pdf*. For example, if your net id is `bondj`, then the file name should be *bondj-HW2.pdf*. HW's submitted in incorrect format (non pdf, incorrect file name etc) will incur a penalty of 30%**