Seoul National University

M1522.000900 Data Structure

Fall 2017, Kang

Homework 2: Algorithm Analysis (Chapter 3)

Due: September 28, 02:00 PM

## Reminders

- The points of this homework add up to 100.
- Like all homeworks, this has to be done individually.
- Lead T.A.: Jungi Jang (elnino9158@gmail.com)
- Please type your answers in English. Illegible handwriting may get no points, at the discretion of the graders.
- If you have a question about assignments, please upload your question in eTL.
- If you want to use slipdays or consider late submission with penalties, please note that you are allowed one week to submit your assignment after the due date.

Remember that:

- Whenever you are making an assumption, please state it clearly

## Question 1

Rank the following expressions by growth rate from fastest to slowest. [20 points]

$$n! \qquad \log_2 n \qquad 2^{10} \qquad 6\mathrm{n}^2 \qquad \log_2 \log_2 n \qquad 3 \cdot 2^{\mathrm{n}} \qquad 20\mathrm{n}$$

## Question 2

Suppose that a particular algorithm has time complexity $\boldsymbol{T}(n) = \boldsymbol{T}(n/2) + 1$ in all cases and that executing an implementation of the algorithm on a particular machine $X$ takes $t$ seconds for $n$ inputs. Answer the following questions. [20 points]

(1) What is time complexity of this algorithm in asymptotic terms? [10 points]

텍스트

(2) Given a new machine $Y$ 4 times faster than $X$, how many inputs could we process in $t$ seconds in $Y$? Express your answer using $n$. [10 points]

텍스트 n

## Question 3

Determine big-Oh for the following code fragments in the average case. Assume that all variables are type of **int**. Be sure to get the tightest bounds for big-Oh. [20 points]

(1) [5 points]

```
sum = 0;
for(i=0; i<n*n;i++)
    sum++;
```

(2) [5 points]

```
sum = 0;
for (i=1; i<=n; i*=2)
    for(j=1;j<=n;j++)
        sum++
```

(3) [5 points]

```
sum = 0;
for(i=1;i<=n;i++) {
    j = n;
    while (i * i < j) {
        sum++;
        j = j - 1;
    }
}
```

(4) Assume that array **A** contains $n$ values, DSutil.random($n$) takes constant time, and **sort** takes $n \log n$ steps. [5 points]

```
for (i=0;i<n;i++) {
    for (j=0;j<n;j++)
       A[j] = DSutil.random(n);
    sort(A);
}
```

## Question 4

Using the definitions of big-Oh and Ω, find the upper and lower bounds for the following expressions. Be sure to get the tightest bounds for big-Oh and Ω. [20 points]

(1) $c_1 n\sqrt{n}$ [5 points]

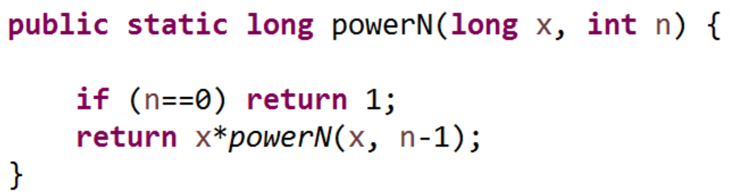
(2) $c_2 n^2 + c_3 n$ [5 points]

(3) $c_4 n \log n^2 + c_5 n$ [5 points]

(4) $c_6 2^n + c_7 n^4$ [5 points]

## Question 5

A power function is a function of the form $f(x) = x^n$. The figures below shows an implementation of the algorithm in Java. Answer the following questions. [20 points]

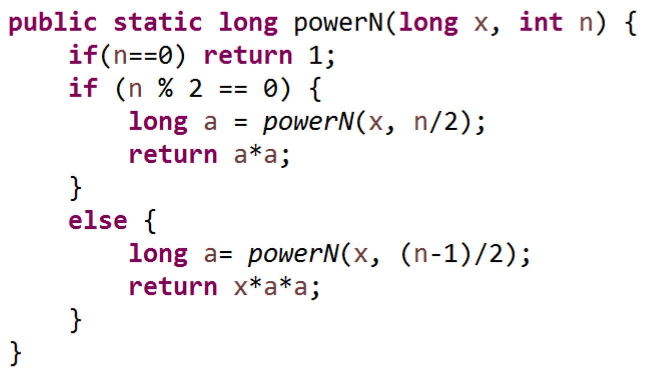


```java
public static long powerN(long x, int n) {

    if (n==0) return 1;
    return x*powerN(x, n-1);
}
```

**Figure 1. A power function implementation in Java.**

(1) In **Figure 1**, what will the time complexity of the power function be in asymptotic terms? Express your answer using $n$. [5 points]



```java
public static long powerN(long x, int n) {
    if(n==0) return 1;
    if (n % 2 == 0) {
        long a = powerN(x, n/2);
        return a*a;
    }
    else {
        long a= powerN(x, (n-1)/2);
        return x*a*a;
    }
}
```

**Figure 2. A modified power function implementation in Java.**

(2) If we change the implementation of the power function from **Figure 1** to **Figure 2**, what will the time complexity be in asymptotic terms? Express your answer using $n$. [15 points]