

Part B – written (Submit as an pdf file, or hand-in in class.)

1. Order following function by growth rate:  $N$ ,  $\sqrt{N}$ ,  $N^{1.5}$ ,  $N \log(N)$ ,  $\log(\log(N))$ ,  $\log(N) \log(N)$ ,  $N^2$ ,  $2^N$ ,  $200$ ,  $N^N$

$$200 < \log(\log(N)) < \log(N) \log(N) < \sqrt{N} < N < N \log(N) < N^{1.5} < N^2 < 2^N < N^N$$

2. Give a useful  $\Theta$  (big Theta) estimation for each of following function  $t(n)$ .

a.  $t(n) = 12^2 * 2^{12}$

$$T(n) = O(1)$$

b.  $t(n) = 2\log_2(n^2) + \log_4(n) + (\log_2 n)^2 + (\log_2(20^2))^2$

$$T(n) = O(\log_2 n^2)$$

c.  $t(n) = 3t(n/2) + n$

$$T(n) = O(n)$$

d.  $t(n) = 3t(n/2) + (n+1)(n-1)$

$$T(n) = O(n^2)$$

e.  $t(n) = 4t(n/2) + (n^2 + n - 1)$

$$T(n) = O(n^2)$$

f.  $t(n)$  is the runtime of following function,

```
public static int f1(int n){
```

```
    int mid = n/2;
```

```
    for (int i = mid; i >= 0; i--) System.out.println(i);
```

$$T(n) = O(n)$$

```
    for (int i = mid + 1; i <= n; i++) System.out.println(i);
```

```
    return mid;
```

```
}
```

g.  $t(n)$  is the runtime of following function,

```
public static int f2(int n){
```

```
    if (n < 1) return 1; //update from original
```

```
    int mid = n/2;
```

```
    mid = f2(mid);
```

$$T(n) = O(\log n)$$

```
    for (int i = 30; i > 0; i /= 3){
```

```
        System.out.println(i);
```

```
    }
```

```
    return mid;
```

```
}
```

h.  $t(n)$  is the runtime of following function,

```
public static int f3(int n){
```

```
    for (int i = n; i >= 0; i--){
```

```
        for (int j = 0, j <= i + i; j++)
```

```
            for (int k = n; k > 0; k /= 3)
```

```
                System.out.println(i * j + k);
```

```
    }
```

```
    return n;
```

```
}
```

$$T(n) = O(n^2 \log_3 n)$$

i.  $t(n)$  is the runtime of following function,

```
public static int f4(int [] a, int start, int end){
```

```
    int ans = 0;
```

```
    if (start >= end) ans = a[start];
```

```
    else {
```

```
        int mid = (start + end) / 2;
```

```
        int x = f4(a, start, mid);
```

```
        int y = f4(a, mid + 1, end);
```

```
        print(a, start, end); //print each element in a from start to end
```

```
        if (x < y) ans = x;
```

```
        else ans = y;
```

```
    }
```

```
    return ans;
```

```
}
```

```
public static void print(int [] a, int s, int e){
```

```
    for (int i = s; i <= e; i++) System.out.println(i);
```

```
}
```

$$T(n) = (n \log n)$$

j.  $t(n)$  the run time of the following method: The method removeLast for a doubly linked list that has size  $n$

$$t(n) = O(n)$$