

## 알고리즘 H/W #1

Due: 10/10(수) 1:50pm

제출: 302동 313-2호

조교: 엄승현

1. (각 5점) Asymptotic running time을 구하고 과정을 밝히시오. Master Thm을 이용할 수 있는 곳에는 이용해도 좋음. Master Thm을 이용하지 않는 경우에는  $O()$ 와  $\Omega()$  모두 구하시오.

1.1  $T(n) = 4T(n/4) + b$   $b$ : a constant

1.2  $T(n) = 3T(n/3) + n \log n$

1.3  $T(n) = 5T(n/5) + 3n$

1.4  $T(n) = T(n/4) + T(3n/4) + \Theta(n)$

1.5  $T(n) = 3T(n/3+9) + n$

2. (20점) 아래 알고리즘의 수행시간을 구하라.

```
sample(A[ ], p, r)
{
    if (r-p≤1) return 1;
    sum = 0;
    for i = p to r
        sum = sum + A[i];
    q = ⌊ (r - p + 1)/3 ⌋;
    tmp = sum + sample(A, p, p+q-1) + sample(A, p+2q, r);
    return tmp;
}
```

3. (25점) 아래 알고리즘 test(n)의 수행 시간의 upper bound를 asymptotic notation으로 나타내어라. n은 양의 정수다. 문제를 풀면서 절대  $\frac{n}{3}+5$ 과  $\frac{2n}{3}+7$ 를  $\frac{n}{3}$ 으로 근사 취급해서는 안된다.

```
int test(n)
{
    if (n ≤ 50) then return n ;
    else return (test( $\frac{n}{3}+5$ ) + test( $\frac{2n}{3}+7$ )) ;
}
```

4. (20점) Worst-case linear-time selection 알고리즘에서 5개로 나누는 대신 3개로 나누면 수행시간은 어떻게 되는가? 결과를 증명하라.
5. (20점) Mergesort에서 둘로 나누는 대신 16 개로 나누어도 sorting은 된다. 즉, 최상위 레벨에서 mergesort를 16 번 부른 다음 merge를 한다. 이 경우의 알고리즘을 기술하고 (상식적인 선에서 기술. 너무 자세할 필요 없음.) complexity를 분석하라. Heap을 이용해서 merge 부분을 효율적으로 하는 방법도 기술에 포함시킬 것.