알고리즘 중간고사

2005. 11. 2 (수요일), open book, 70분

1. (15점) Asymptotic running time을 밝혀라. (Upper bound만 밝힐 것. 과정 포함)
$$T(n) = T(\alpha n) + T(\beta n) + O(n), \qquad \alpha, \beta > 0, \quad 0.9 < \alpha + \beta < 1$$

2. (21점) 이미 오름순으로 sorting 되어 있는 입력에 대해 다음의 algorithm을 사용해서 오름순 sorting을 하면 소요되는 asymptotic time은 각각 어떻게 되는가?
   A. Quicksort
   B. Mergesort
   C. Min-heap을 사용하는 Heapsort

3. (15점) 아래 알고리즘의 asymptotic running time을 구하려면 당신이라면 어떤 것을 기준으로 삼겠는가? 이에 기반하여 sample(A, 1, n)을 수행할 경우 worst-case asymptotic running time의 upper bound를 구하라. (최대한 tight하게) 분석 능력을 묻기 위해 만든 문제이니, 알고리즘이 실제로 어떤 의미있는 일을 하는지는 신경쓰지 말 것. $n = 5^k$ (k는 자연수)이라고 가정할 것.

```
algorithm   sample( A, p, r)
// A[p … r]:   given array
{
      if ( r - p > 4 )
      {
            q = ( r – p + 1 ) / 5;
            sample( A, p, p + 2*q – 1 );
            sample( A, p + 2*q, p + 3*q – 1 );
            sample( A, p + 3*q, r );
      }
}
```

4. (15점) 수업시간에 배운 partition을 사용해서 quicksort를 할 경우 quicksort가 stable sort가 되는가? 이유를 간단히 설명하라.

5. Radix sort가 제대로 sorting한다는 것을 귀납적으로 증명하라.

6. (20점) 집합 {0, 1, ⋯, 9 }의 원소들이 n 개 늘어선 수열이 있다. 역시 같은 집합의 원소들이 m개 늘어선 또다른 수열이 주어진다. 이 두 수열을 가능한 한 점수가 높은 방식으로 짝을 맞추려 한다. 같은 자리에 맞추어진 두 수가 같으면 그 수만큼 더해준다. 같은 자리에 맞추어진 두 수가 다르면 두 수의 차이의 절대값 만큼 빼준다. 짝이 없는 수 각각에 대해서는 2를 빼준다. 예를 들어, 250863459, 68241 두 수열에 대해서 아래와 같이 짝을 맞추면 ('-'는 짝이 없는 경우를 나타냄)

   250863459
   -6-8-241-

   점수는 -2-1-2+8-2-1+4-4-2 = -2 가 된다. 긴 수열에는 절대 공백이 있으면 안 되고, 짧은 수열에는 적어도 하나 이상의 공백이 생긴다. 각 수열에서 원소들의 순서가 바뀌면 안 된다. 길이가 각각 n, m $(n > m)$인 두 수열을 위와 같이 맞출 때 가능한 가장 높은 점수를 알아내는 dynamic programming을 제시하고 (점수만 알아내면 됨), 당신이 제시한 방법의 asymptotic running time을 밝히라. (dynamic programming은 관계식만 나오면 됨.) 채점상의 불이익을 피하기 위해서, 변수를 정의할 때 변수의 의미를 정확히 밝힌 다음 식을 전개해야 한다.