

## Computer Algorithms, 기말고사

2011. 12. 12  
Open book, 70분

\* 각 20점. 혹시 문제를 푸는데 가정이 필요하면 직접 할 것. (출제 과정에서 미처 생각지 못한 부분이 있을 경우에 해당됨.) 가정의 합리성은 생각의 성숙도를 나타내는 것이므로 남용하면 안됨. 출제자가 미처 생각지 못한 부분을 반영하는 가정은 가산점 부여함.

1. 아래는 수업 시간에 배운 알고리즘 strongly-connected-components( )가 제대로 작동한다는 것에 대한 증명 과정의 일부다.

"( $\leftarrow$ ) Assume that vertices  $v$  &  $w$  are in the same tree in the DFS forest of  $G^R$ . There is a path from the root  $x$  to  $v$  in  $G^R$ . This means that there is a path from  $v$  to  $x$  in  $G$ . Suppose to the contrary that there is no path from  $x$  to  $v$  in  $G$ . Then  $f(x) < f(v)$ . This is a contradiction, since  $f(x) > f(v)$  by Step 3.(1). Similarly, we can show ..."

여기서 밑줄친 ① 부분을 왜 이렇게 말할 수 있는지 설명하라.

2. 아래는 Shortest Path를 구하기 위한 Dijkstra 알고리즘이다. 이는 시작 vertex 하나로부터 모든 다른 vertex로 가는 최단경로를 다 구한다. 자동차 네비게이터에서는 지도가 그래프로 주어지고 시작점에서 특정한 목적점까지의 최단경로를 구하는 작업을 한다. 이 작업을 위해 Dijkstra 알고리즘을 사용하는 것은 적당하지 않다.

2.1 (5점) Dijkstra 알고리즘을 사용하는 것이 왜 비효율적인가?

2.2 (15점) 이를 개선하기 위한 아이디어를 내어 보라. 정답까지 도달하기는 쉽지 않으니 자신의 생각이 미친 곳까지 최선을 다해 기술할 것.

```
Dijkstra( $G, r$ )
▷  $G = (V, E)$ : 주어진 그래프
▷  $r$ : 시작으로 삼을 vertex
{
     $S \leftarrow \phi$ ;                                ▷  $S$ : vertex 집합
    for each  $u \in V$ 
         $d[u] \leftarrow \infty$ ;
     $d[r] \leftarrow 0$ ;
    while ( $S \neq V$ ) {
         $u \leftarrow \text{extractMin}(V - S, d)$ ;
         $S \leftarrow S \cup \{u\}$ ;
        for each  $v \in L(u) \cap (V - S)$  ▷  $L(u)$ :  $u$ 의 adjacent vertex 집합
            if ( $d[v] > d[u] + w(u, v)$ )
                then {
```

```

        }
    }
    extractMin(Q, d)
    {
        집합 Q에서 d 값이 가장 작은 vertex u를 리턴한다;
    }

```

3. NP-Complete, NP-Hard 이론에 관한 다음 물음에 답하라.

3.1 (5점) NP가 아니면서 NP-Hard인 문제가 있을 수 있는가? 대답에 대한 이유도 말해야 한다.

3.2 (15점) 3SAT 문제는 수업 시간에 배운대로 다음과 같이 정의된다.

$3SAT = \{ \langle \phi \rangle \mid \phi \text{ is a satisfiable Boolean formula in 3-CNF} \}$

MAX-3SAT 문제는 3-CNF 형태인 임의의 Boolean formula가 주어졌을 때 satisfiable clause의 최대 개수를 알아내는 optimization 문제다. 3SAT이 NP-Complete란 사실을 이용하여 MAX-3SAT이 NP-Hard임을 증명하라.

4.  $n \times n$  문자 배열에서  $m \times m$  패턴을 찾을 수 있도록 Rabin-Karp 알고리즘을 확장하는 방법을 설명하라. 최대한 효율적으로 확장할 것. 알고리즘을 다 쓸 필요는 없고 핵심 부분을 어떻게 바꾸면 될지를 명확히 설명하면 된다. 당신이 제안한 방법의 asymptotic complexity는 어떻게 되는가? Asymptotic complexity를  $n$ 과  $m$ 에 대해서 구하라. (반드시  $m \ll n$  이라고 가정하지 말 것.  $m < n$  이라고만 말할 수 있음.  $m$ 이 아주 커질 수도 있다.)

5. Floyd-Warshall 알고리즘에 대해 다음에 답하라.

5.1 (5점) Shortest path 문제에서 음의 cycle은 아예 허용하지 않는다. 만일 음의 cycle이 존재한다면 Floyd-Warshall 알고리즘이 수행된 후, 그런 cycle이 존재한다는 것을 어떻게 알 수 있는가?

5.2 (15점) Floyd-Warshall 알고리즘에서 intermediate vertex 집합을 잡는 데 있어 vertex의 순서는 아무렇게나 되어도 상관없다. 즉, vertex의 index에 상관없이 Floyd-Warshall 알고리즘은 바르게 작동한다. 이를 어떻게 증명하면 될 것인지 생각해 보아라. 여러분에게 증명을 완성하길 원하지 않는다. 출제 교수도 생각해보니 정답을 완성하기는 쉽지 않다. 생각이 어디까지 어떻게 전개되는지를 보려한다.