

Dec. 10, 2014. Open book. 65분

* 그냥 단답형 대답은 모든 문제에서 점수를 받을 수 없음. 어떤 경우든 이유를 설명해야 함. 그런 일은 없을 거라 생각하지만 만일 불확실한 것이 있다면 본인이 직접 가정을 하고 진행할 것. 합리적인 가정을 하는 것도 실력임.

1. (20점. 각 5점) 다음 각각에 대해 O 또는 X로 대답하라.

(O, X 공히 이유를 설명해야 한다.)

- a. NP가 아니라도 NP-Complete에 속할 수 있다.
- b. $P \subseteq \text{CO-NP}$
- c. 최적화 버전의 TSP는 “NP-hard”다.(이것은 사실임) 만일 이 문제가 polynomial-time에 풀리면 모든 “NP-complete” 문제들이 polynomial-time에 풀린다.
- d. 모든 edge의 weight이 1부터 50 사이의 정수라면, minimum spanning tree를 구하는 두 알고리즘 Prim과 Kruskal 중 Kruskal이 더 빠르다.

2. (20점) 철수가 어떤 그래프에 대해 minimum spanning tree를 구하였는데, 조교가 단 한 개의 edge가 틀렸다고 말했다. 조교가 틀린 edge도 알려주었다. 따라서 이 틀린 edge는 일단 제거했다. 이 상태에서 맞는 minimum spanning tree를 찾는 가능한 한 빠른 알고리즘을 제시하고, 그 시간복잡도를 밝히고 시간복잡도가 왜 그렇게 되는지 설명하라. 이 시간이 그냥 처음부터 minimum spanning tree를 구하는 것보다는 당연히 빨리 끝나야 한다.

(참고: 어떤 tree에서 traversal은 $\Theta(|V|+|E|)$ 시간이 소요된다.)

3. (20점) DFS의 끝에 아래와 같이 finishing time($f_time[v]$)을 표시하도록 하였다.

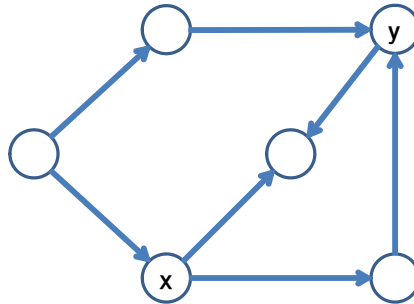
```

bDFS(v)
{
    mark[v] = YES;
    for all x ∈ L(v) // L(v): vertex v에 adjacent한 vertex 집합
        if (mark[x] = NO) bDFS(x);
    f_time[v] = cnt++;
}

```

입력 그래프가 주어지면, 임의의 vertex에서 위의 bDFS를 시작하여 더 이상 진행할 수 없을 때까지 진행한다. 만일 이 bDFS가 모든 vertex를 다 방문하지 못하고 끝나면 나머지 vertex들 중 아무거나 하나 골라 다시 bDFS를 수행한다. 이런 식으로 모든 vertex가 방문될 때까지 진행한다. 이 문제에서의 DFS란 이 알고리즘을 말하기로 하자. 이 과정에서 변수값 cnt는 일관성을 갖고 계산된다.(즉, global variable처럼 동작한다.)

아래 그림으로 위의 DFS를 수행했을 때, vertex x의 finishing time이 vertex y의 finishing time보다 빠른 경우가 있는가? bDFS는 아무 vertex에서나 시작할 수 있다는 점을 반드시 고려해야 한다.



4. (20점) String matching을 위한 Rabin-Karp 알고리즘에서 영문 소문자로만 제한된 text 만을 대상으로 하는 경우를 가정하자. Text의 모든 n 개 symbol이 'a'로 채워져 있고(즉, "aaaaaaaaaaaaaaaaa..."), 찾고자 하는 패턴으로 길이 $n/5$ 인 "aa...aa"가 입력될 때 Rabin-Karp 알고리즘의 수행시간은 어떻게 되는가? $\Theta(\)$ -notation으로 나타내라. 시간의 근거에 대한 간단한 설명 붙일 것. (첫번째 매칭만 찾고 끝나는 것으로 착각하지 말 것)
5. (20점, 각 10점) "HAM-CYCLE" 문제와 "두 점 사이 HAM-PATH" 문제는 아래와 같이 정의된다.

HAM-CYCLE = $\{ \langle G \rangle \mid \text{undirected graph } G=(V, E) \text{에 hamiltonian cycle이 존재한다} \}$
 두 점 사이 HAM-PATH = $\{ \langle G, x, y \rangle \mid \text{undirected graph } G=(V, E), x, y \in V, \text{ vertex } x \text{에서 } y \text{에 이르는 hamiltonian path가 존재한다} \}$

- * Hamiltonian path: 그래프의 모든 vertex가 한번씩 나타나는 경로
- * Hamiltonian cycle: 그래프의 모든 vertex가 한번씩 나타나는 사이클

이 두 문제 사이에는 어느 하나가 NP-hard이면 다른 것도 NP-hard가 된다. 즉, 서로 간에 양방향으로 poly-time reduction이 가능하다. 이것을 각각 해보라. 즉, 아래를 보이라.

5.1 두 점 사이 HAM-PATH \leq_p HAM-CYCLE

5.2 HAM-CYCLE \leq_p 두 점 사이 HAM-PATH