2020 알고리즘 Prof.조승범 Task (20.12.2)					
학과	소프트웨어	학번	2019038004	이름	조민우

1. Adversary argument를 이용하여 size n인 array에서 maximum element를 찾는데 최소 n-1 번의 comparison이 필요함을 증명하시오.

Array의 Size = n
Adversary에 의해 A[i] > A[j] 일때, i는 j에 대해 이기고, j는 i에 대해 졌다고 정의한후, Array index set을
W : 한번 이상 이기고, 지지 않은 index number들의 집합
L : 한번 이상 진 index number들의 집합
N : 한번도 비교하지 않은 index number들의 집합
로 분할한다.
초기 상태는 N = [i for i in 0 to n-1], L = [], W = [] 가 된다.
이때 N에 모든 요소가 다 제거되었을때 W에 한가지 요소만이 존재한다면 W에 있는 요소가 Maximum element의 index number이다.
N의 size가 1 줄거나 L의 size가 1 증가할때 1 credit을 획득한다.
이때 Problem을 Solve하기 위해 필요한 최소 credit은 N의 1개의 요소가 W로 이동 -> 1, 남은 N의 n-1개의 요소가 L로 이동 -> 2(n-1)이므로, 총합으로 최소 2n-1개의 credit이 필요하다.

(i,j)	A[i]와 A[j]를 비교한 후 L과 N의 size 변화	획득하는 최소 Credit
(N,N)	len(L) += 1, len(N) -= 2	3
(N,W) or (W,N)	len(L) += 1, len(N) -= 1	2
(N,L) or (L,N)	len(L) += 1 or 0, len(N) -= 1	1
(W,W)	len(L) += 1	1
(L,L)	Not change	0
(W,L) or (L,W)	len(L) += 1 or 0	0

(대각선이 그어진것은 비교했을 시에 Credit의 변화가 없어서 무의미하다)

이를 최소 Credit이 큰 순서대로 Greedy하게 뽑기 전에, n = 2k일 때와 n = 2k + 1일 때로 case를 나눈다.

Case n = 2k) least need Credit = 2n − 1 = 4k − 1
Default len(N) = 2k, len(L) = 0, len(W) = 0
(N, N) 비교를 최대 k번 가능하다. -> 3k credit earned, total = 3k
이때 len(N) = 0, len(L) = k, len(W) = k
len(N) == 0이므로, N이 들어간 비교는 할 수 없다.
(W, W) 비교를 최대 k - 1번 가능하다. -> k - 1 credit earned, total = 4k - 1
이때 len(N) = 0, len(L) = 2k - 1, len(W) = 1
∴ 총 반복 횟수는 k + k − 1 = 2k − 1 = n − 1

Case n = 2k + 1) least need Credit = 2n − 1 = 4k + 1
Default len(N) = 2k + 1, len(L) = 0, len(W) = 0
(N, N) 비교를 최대 k번 가능하다. -> 3k credit earned, total = 3k
이때 len(N) = 1, len(L) = k, len(W) = k
(N, W) 혹은 (W, N) 비교를 최대 1번 가능하다. -> 2 credit earned, total 3k + 2
이때 len(N) = 0, len(L) = k + 1, len(W) = k
(W, W) 비교를 최대 k - 1번 가능하다 -> k - 1 credit earned, total = 4k + 1
이때 len(N) = 0, len(L) = 2k, len(W) = 1
∴ 총 반복 횟수는 k + 1 + k − 1 = 2k = n − 1

2가지 Case에 의해서, size n인 array에서 maximum element를 찾는데 최소 n-1 번의 comparison이 필요함을 증명할 수 있다.

2. 문제 X, Y, Z 에 대해 X $\le_p$ Y 이고, Y $\le_p$ Z 이면 X $\le_p$ Z 임을 증명하시오.

X와 Y사이의 Reduction을 R_1, Y와 Z사이의 Reduction을 R_2라고 했을 때,
R_1을 이용해 X의 Instance I_X를 Y의 Instance I_Y로 변환할 수 있다.
또, R_2를 이용해 Y의 Instance I_Y를 Z의 Instance I_Z로 변환할 수 있다.
이때, $X \le_p Z$ 가 성립하지 않는다고 가정하면, X의 Instance 중 하나인 α를 R_1을 통해 β로 변환하였을 때, β에 대한 R_2의 변환 결과가 Z의 Instance인 I_Z가 아니어야 한다. 이는 R_2의 정의에 모순이되므로, $X \le_p Z$ 가 성립함을 증명할 수 있다.

3. Independent Set problem이 NP-complete라는 사실을 이용하여 clique problem이 NP-complete임을 증명하시오.

Independent Set $\leq_p$Clique 임에서 Cook-levin theorem에 의해 Clique Problem의 NP-hardness는 증명할 수 있다.
이제, NP-Completeness를 증명하기 위하여, Clique Problem이 NP에 속해있음을 증명하면 된다.

Clique Problem
Def) 어떤 Graph G가 size가 k 이상인 Clique을 가지고 있는지 판별
Certificate) 어떤 set $S \subset V$
Certifier) $|S|$가 k 이상인지 확인 후, S에 속한 모든 vertex가 서로 adjacent한지 확인

어떤 instance s(G, k)에 대해 Certificate t와 Certifier C(s, t)를 위와 같이 정의한다.
이때, t의 size는 O(n)이고, s의 size는 O(n+m)이기 때문에 t의 size는 $|s|$보다 작을 수 밖에 없고, $|s|$에 대한 polynomial로 표현된다.

case S is YES instance) 일때, 정의에 의해 C(s, t)인 Certificate t가 존재한다.
else case) 모든 string t에 대해서 C(s, t)는 No를 return한다.

C(s, t)는 모든 vertex pair(u, v)에 대해 adjacent 한지를 검사해야 하므로, O($n+n^2$)의 시간이 소모되고, 이는 O($poly(n)$)으로 표현되어 C가 efficient certifier임을 증명할 수 있다.

그러므로 Clique Problem은 Efficient Certifier을 가지므로 NP가 되고, NP-hardness는 이미 증명되었으므로 NP-Complete임을 알 수 있다.

4. Consider the CLIQUE problem restricted to graphs in which every vertex has degree at most 3. Call this problem CLIQUE-3.
(a) Prove that CLIQUE-3 is in NP.
(b) What is wrong with the following proof of NP-completeness for CLIQUE-3?
We know that the CLIQUE problem in general graphs is NP-complete, so it is enough to present a reduction from CLIQUE-3 to CLIQUE. Given a graph G with vertices of degree 3, and a parameter g, the reduction leaves the graph and the parameter unchanged: clearly the output of the reduction is a possible input for the CLIQUE problem. Furthermore, the answer to both problems is identical. This proves the correctness of the reduction and, therefore, the NP-completeness of CLIQUE-3.
(c) It is true that the VERTEX COVER problem remains NP-complete even when restricted to graphs in which every vertex has degree at most 3. Call this problem VC-3. What is wrong with the following proof of NP-completeness for CLIQUE-3?
We present a reduction from VC-3 to CLIQUE-3. Given a graph $G=(V,E)$ with node degrees bounded by 3, and a parameter b, we create an instance of CLIQUE-3 by leaving the graph unchanged and switching the parameter to $|V|-b$. Now, a subset $C \subseteq V$ is a vertex cover in G if and only if the complementary set $V-C$ is a clique in G. Therefore G has a vertex cover of size$\leq b$ if and only if it has a clique of size $\geq |V|-b$. This proves the correctness of the reduction and, consequently, the NP-completeness of CLIQUE-3.
(d) Describe an O($|V|^4$) algorithm for CLIQUE-3.

(a) Clique-3 Problem
Def) 모든 Vertex의 degree가 최대 3인 어떤 Graph G가 size가 k 이상인 Clique을 가지고 있는지 판별
Certificate) 어떤 set $S \subset V$
Certifier) $|S|$가 k 이상인지 확인 후, S에 속한 모든 vertex가 서로 adjacent한지 확인

어떤 instance s(G, k)에 대해 Certificate t와 Certifier C(s, t)를 위와 같이 정의한다.
이때, t의 size는 O(n)이고, s의 size는 O(n+m)이기 때문에 t의 size는 |s|보다 작을 수 밖에 없고, |s|에 대한 polynomial로 표현된다.

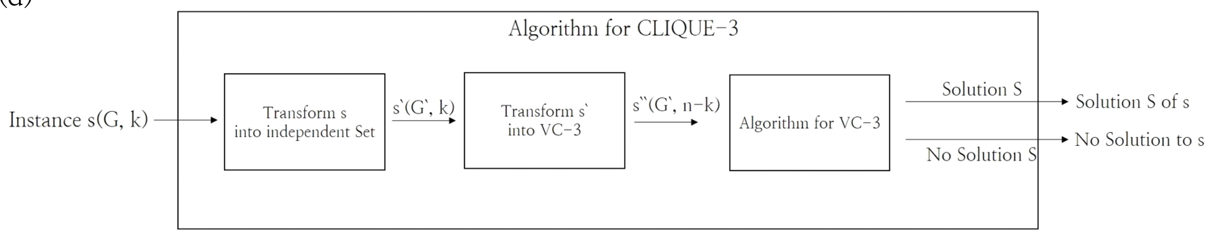
case S is YES instance) 일때, 정의에 의해 C(s, t)인 Certificate t가 존재한다.
else case) 모든 string t에 대해서 C(s, t)는 No를 return한다.

C(s, t)는 모든 vertex pair(u, v)에 대해 adjacent 한지를 검사해야 하므로, $O(n+n^3)$의 시간이 소모되고, 이는 $O(poly(n))$으로 표현되어 C가 efficient certifier임을 증명할 수 있다.

그러므로 Clique-3는 NP이다.

(b) 처음 reduction을 가정할 때, reduction의 방향이 잘못되었다. 이유는 간단하다. Clique-3는 모든 vertex의 degree가 최대 3인 그래프라는 조건이 주어져 있으나, Clique는 이를 포함하는 모든 vertex에 대해 NP이므로, 실질적으로 Clique-3 $\leq_p$ Clique임을 알 수 있다.

(c) Vertex Cover Problem에서 Clique Problem으로의 관계가 증명되지 않았다. 이를 수정한다면 Vertex Cover -> Independent Set -> Clique의 과정을 거쳐야한다.

(d)


#Instance에 대한 Solution이 전부 같은 값을 반환하므로 Solution에 대한 변환은 필요 없다.