2019038004 조민우

"저는 이 시험에서 부정행위가 발각시 어떠한 조치를 취하더라도 해당 조치에 모두 따르겠습니다." ~~(인)~~

1. 목적 수열 → internal node n, leaf node $n+1$

base case : internal node 0

→ 당연하게도 leaf node는 1개이다. (node가 1개 밖에 없음)

if) internal node $k-1$ 일때 leaf node가 k 가 성립한다 가정

if) internal node k, leaf node가 $k+1$ 이 아니라면,

Huffman tree의 기본적인 정의인 full binary tree가 성립하지 않는다.

(노드가 1개 증가해야함)
물론

∴ internal node가 n 이면 leaf node는 $n+1$개임을 증명

2.

(1) LCS : Longest Common Subsequence.

문자열 X와 Y의 subsequence 중, 공통되는 subsequence 중에서 가장 긴 수열을 구하는 문제.
(연속될필요X, 순서지켜야함)

(2)

$$LCS(i,j) = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ max(LCS[i-1][j], LCS[i][j-1]) & \text{if } S1[i] != S1[j] \\ LCS[i-1][j-1]+1 & \text{if } S1[i]=S2[j] \end{cases}$$

3. 성립하지 않는다.

∵ 위 recurrence relation은, 이전의 MAS가 음수일때(a.k.a 0보다 작을때) 현재의 값을 MAS(i)에 대입한다.

그러나, / 2 -1789 / 이란 반례를 보면

$i=0$ $MAS(0)=0$

$i=1$ $MAS(1)=1$

$i=2$ $MAS(2)=3$

$i=3$ $MAS(3)=-1786$

$i=4$ $MAS(4)=1$

이라는 결과를 만들어 MAS(i)가 maximum subarray를 구성하는 element들의 총합이 아님을 볼수있다.

이 식을 성립시키려면 "A[i]를 포함하는 A[1 ... i]의 maximum subarray를 구성하는 element 들의 총합" 으로 정의해야한다.

4. 어떤 i에 대해서 MAS(i)를 저장해야 하므로 1차원 array를 data structure로 사용한다.

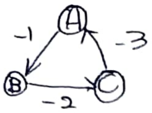
이때 MAS(i)를 구하는데 MAS(i-1)이 필요하므로 Subproblem(i)는 Subproblem(i-1)에 의존한다.

5. 어떤 int type value를 선언하고 (예시 m)

```
m = MAS(1)
for i 2 to n :
    m = max(m, MAS(i))
```

와 같이 모든요소들 중 최대값을 구하면된다.

6. negative cycle : weight의 합이 0 미만인 cycle.



→ 알고리즘을 동작시키면 계속해서 update되므로 결과를 내지 못함.

7. Counting Sort's time complexity : $O(n)$.
일반적인 sorting은 comparison-based sorting을 하지만 Counting sort는 non-comparison sort이기 때문이다.

8. 3-SAT의 각 clause에 literal을 한개씩 추가한다면, 우린 4-SAT을 ~~환원~~할 수 있다.
: 다시 생각한다면, 4-SAT의 각 clause에서 literal을 하나씩 제거한다면 3-SAT으로 ~~변형~~할 수 있다.
, 4-SAT의 각 clause에서 제거할 수 있는 모든 경우에 대해 3-SAT을 만들어 satisfiability를 검사하고,
다시 literal을 추가하여 검사한다면 4-SAT을 해결할 수 있다.
3-SAT을 해결하는 알고리즘이 polynomial time 안에 4-SAT도 해결할 수 있으므로

$$3\text{-SAT} \le_p 4\text{-SAT}.$$

9. NP-Hardness는 증명되었으므로 (cook-levin), 4-SAT이 NP임을 증명하면된다.
Problem : 어떤 4-CNF formula F는 satisfiable한가?
1) Certificate.
: F의 각각의 variable에 0/1을 assign한것.
2) Certifier.
: 해당 assignment에 대해 F가 true인지 판별.
instance s에 대해, certificate t와 certifier $C(s,t)$를 정의.
s가 YES instance라면, 정의에 의해 $C(s,t)$인 t의 존재가 자명함.
$C(s,t)$는 t에 속한 각 clause에 대해 $O(1)$안에 대입 검증이 가능하므로, $O(|t|)$
∴ $C(s,t)$ = efficient certifier

→ 4-SAT ∈ NP.

NP-Hardness와 NP에 속함이 증명되었으므로,
4-SAT은 NP-Complete이다.