"야유회 (workshop)" 문제 풀이

작성자: 구재현

부분문제 1

```
return my_num;
```

17점 ($m = 34$)

`afternoon`, `evening` 함수는 `return my_num;` 을 하게 하고, `morning` 함수만 사용한다.

모든 $0 \le u, v, w \le N-1, u \neq v, v \neq w$ 에 대해서 $f(u, v) \neq f(v, w)$ 를 만족하며 반환값이 충분히 작은 함수 f를 구성하자. 이러한 함수가 있으면, `morning` 함수는 단순히 $f(my\_num, right\_num)$ 을 반환하면 된다.

u 와 v 를 이진 비트열이라고 생각하자. $u \neq v$이기 때문에 둘이 다른 값을 가지는 비트가 존재할 것이다. $u[k] \neq v[k]$ 라고 할 경우, $f(u, v) = 2k + u[k]$ 라고 정의하자. $f(u, v) = f(v, w)$ 일 경우, $u[k] = v[k]$ 가 만족해야 하는데, 이는 가정에 모순이다.

이렇게 구성한 $f(u, v)$ 는 최대 $2\lceil \log_2 N \rceil$ 의 크기를 가진다. 입력 제한에서 이는 최대 34 이며 16점을 얻는다.

46점 ($m = 8$)

`afternoon`, `evening` 함수의 `left_num` 인자를 무시하자. 17점의 방법을 나머지 두 함수에 그대로 적용하면, $34 \to 12, 12 \to 8$ 로 크기를 줄일 수 있다.

57점 ($m = 6$)

`afternoon`, `evening` 함수의 `left_num` 인자를 활용해서 크기를 한번 더 줄일 수 있다. $comp(u, v, T)$ 를, $0 \le u, v \le 2^T - 1, u \neq v$ 인 두 정수가 주어졌을 때 $[0, 2T-1]$ 이하의 정수를 반환하는 함수라고 하자 ($f(u, v)$ 의 정의를 약간 변형하였다). `morning` 함수에서 크기를 34 이하로 줄였다면,

- `afternoon(l, x, r) = comp(comp(l, x, 6), comp(x, r, 6), 4)`
- `evening(l, x, r) = comp(x, r, 3)`

와 같다. 위의 풀이에서 항상 인접한 수가 다르다는 불변량을 유지하고 있기 때문에, 인자가 3개 들어오는 경우 압축을 두 번 수행할 수 있다.

$comp$ 함수는 $[0, 5]$ 범위의 정수를 압축할 수 없기 때문에, `evening` 함수에서는 `left_num` 인자를 참조하여도 도움이 되지 않는다.

81점 ($m = 4$)

현재의 $comp(u, v, T)$ 는 $[0, 2^T - 1]$ 의 수를 받아 $[0, 2T-1]$ 의 수를 반환하는데, 이 때문에 6 미만의 압축이 불가능하다는 한계가 있다. 함수의 정의를 조금 바꾸면 4 까지 압축이 가능하다.

길이 $2T$ 의 이진 문자열 중 0 이 T 개이고 1 이 T 개인 문자는 $\binom{2T}{T}$ 개이다. $0 \le u, v \le \binom{2T}{T} - 1$ 인 두 수 u, v 에 대해서, 이 두 수를 위 성질을 만족하는 서로 다른 이진 문자열로 매핑하자. 1 의 개수가 고정되어

있기 때문에, $u[k] = 0, v[k] = 1$ 인 k 가 무조건 존재한다. 또한, $f(u,v) \neq f(v,w)$ 가 성립할 수밖에 없다 $(v[k] = 1)$. 이를 사용하면. $comp(u, v, T)$ 의 입력 크기 제한을 $[0, 2^T - 1]$ 에서 $[0, \binom{2T}{T} - 1]$ 로 제곱에 가깝게 늘릴 수 있다. 이제

- `morning(x, r) = comp(x, r, 10)`
- `afternoon(l, x, r) = comp(comp(l, x, 3), comp(x, r, 3), 2)`
- `evening(l, x, r) = x`

로 두면 된다 $(\binom{20}{10} \geq 180\,000, \binom{6}{3} = 20, \binom{4}{2} = 6)$.

이 함수 역시 4 미만의 압축을 할 수는 없기 때문에, `evening` 함수는 사용하지 않는다.

이진 문자열로의 매핑은, `init` 함수에서 모든 $\binom{2T}{T}$ 개의 조합을 단순히 다 나열하는 것으로 쉽게 구현할 수 있다.

만점 ($m = 3$)

`evening` 함수에서 $[0, 3]$ 범위의 세 정수 l, x, r 이 주어졌을 때, 이를 $[0, 2]$ 범위의 정수로 매핑해야 한다. 이는 크게 두 가지 방법으로 해결할 수 있다.

가장 간단한 방법은 다음과 같다. 만약 $x \leq 2$ 일 경우, x 를 그대로 반환해 준다. $x = 3$ 일 경우, $l, r \leq 2$ 이기 때문에 이 두 수가 바뀌지 않음이 보장된다. $\{0, 1, 2\}$ 중 $\{l, r\}$ 중 어느 것도 아닌 수가 무조건 하나 존재하기 때문에, x 를 이 수로 바꿔도 인접한 수가 다르다는 조건이 유지된다.

그 외, 위 문제를 36 개의 정점을 가진 그래프의 3-coloring 문제로 변환한 후, 이 값을 로컬에서 백트래킹으로 찾고, 백트래킹 결과를 하드코딩하는 방법도 있다. 잘 구현된 백트래킹은 순식간에 3-coloring을 찾을 수 있다.

여담

이 문제는 Distributed Computing 환경에서 Graph Coloring 문제를 해결하는 것과 밀접한 연관성이 있다. 위 문제의 세팅은 사이클의 노드의 값이 서로 다른 parallel machine에 저장되어 있을 때, $O(\log^*(n))$ 번의 Iteration 만을 사용하여 사이클의 3-coloring을 찾는 것이라고 생각할 수 있다. 여기서 $\log^*(n)$ 은, $n \leftarrow \log_2 n$ 을 반복했을 때 $n \leq 1$ 이 되게끔 하는 최소 반복 수를 뜻한다.

일반적인 그래프에서도 위의 접근을 확장시켜, 그래프의 최대 차수를 Δ 라고 할 때 $(\Delta + 1)$-coloring을 $O(\log^*(n) + \Delta)$ 시간에 찾을 수 있다. 간단히 요약하자면, 위 풀이는 Functional graph에서도 동일하게 적용할 수 있고, 그래프를 Δ 개의 Functional graph로 분해한 후 각각에 대해 위와 같은 알고리즘을 동일하게 적용한 후 이를 합쳐줄 수 있다. 이러한 알고리즘을 *Cole-Vishkin Algorithm* 이라고 한다.

관련된 내용을 더 탐구해 보고 싶으면 `https://jukkasuomela.fi/dda-2010/lecture-2.pdf` 을 읽어보면 좋다.