

< GBC_ Algorithm PA 10 >

27기 최 하영

#백준 1670_ 정상 회담 2

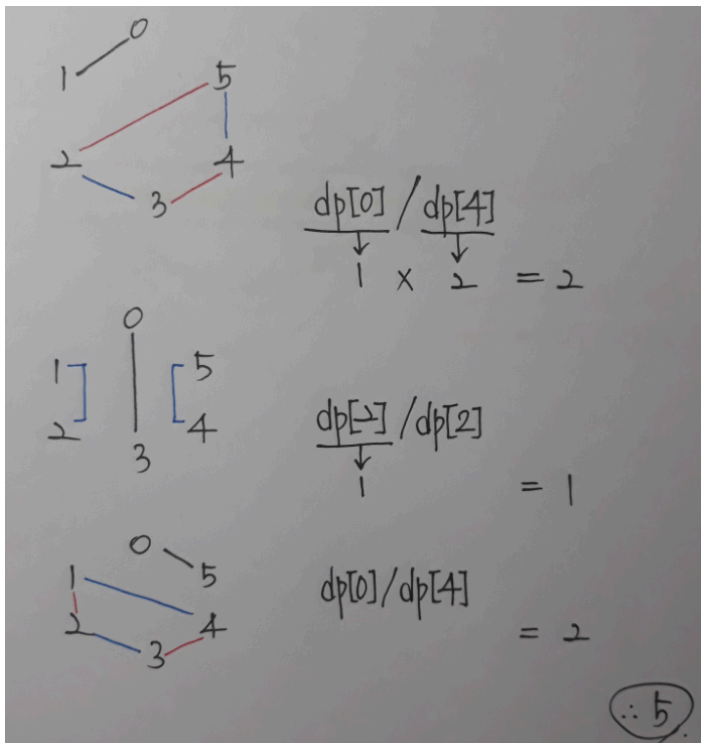
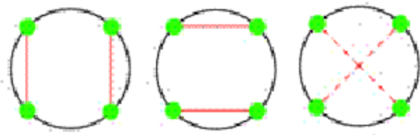
1. Fact

* 원탁의 자리를 배정받은 n 명의 사람들이 팔을 교차하지 않고 완벽하게 악수하는 경우의 수를 구하는 프로그램

* 입력 : 사람의 수 N (10,000 보다 작거나 같은 짝수)

* 출력 : 완벽한 악수의 경우의 수를 987654321로 나눈 나머지

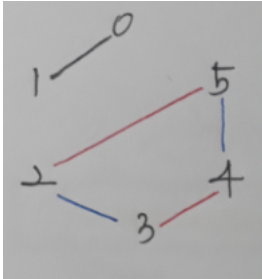
2. Overviews



- 1) 위와 같은 경우로 악수 가능
- 2) 처음에 '완벽하게 악수한다'는 것이 한 명도 빠짐없이 악수를 하고 있어야 한다는 의미로 이해하지 못해 헤맸다. 모든 사람이 악수를 하고 있어야 한다.
- 3) Dp문제이므로, 점화식을 세워 해결하자!

3. Algorithm

- 1) 사람의 수 n 을 입력 받는다.
- 2) $dp[n]$ 을 n 명이 완벽한 악수를 하는 경우의 수라고 한다.
- 3) 시작점인 사람은 짝수 거리로 떨어진 사람들과 악수 가능하다.
→ (= 사람의 수를 2씩 감소시키며 경우의 수를 더해야 한다.)



- 4) 0과 1이 악수를 할 때, 남은 사람들은 반드시 2-5/3-4 이런 식으로, 선을 그었다 했을 때, **같은 라인끼리만 악수**를 해야한다. (교차 X)
- 5) 따라서 각각 경우의 수를 구한 다음, 두 경우의 수를 곱하면, 떨어져 있는 사람과 악수하는 경우의 수를 구할 수 있다.
- 6) 설명이 어려워서 코드를 가져와봤다 .. 아래 점화식을 보면 이해가 될 것이당 ..

```
for(int i=2; i<=n; i+=2)
{
    for(int j=0; j<= i-2; j+=2)
        dp[i] += dp[j] * dp[i-j-2] % tmp;
}
```

- 7) Tmp는 문제에서 말한 987654321을 의미하는 변수다.
- 8) 마지막에 $dp[n]$ 을 출력해주면, n 명이 악수하는 경우의 수가 나온다.
- 9) +) 위 방법 말고 다른 방법으로도 풀어봤으나 그림으로 설명하기에, 이 방법이 제일 직관적이라고 생각했다.

4. Time complexity

$O(n^2)$ → 이중 for 문을 돈다.