

# week4

2024-07-25

# 베르트랑 공준

$N \leq 123456$

$(n, 2 * n]$  사이의 소수 개수 구하기

에라토스테네스의 체에 추가로 누적합까지 사용할 수 있음

# 베르트랑 공준

```
for(int i = 2; i <= 123456 * 2; i++) {  
    psum[i] = psum[i - 1] + (int)p[i];  
}
```

```
while(true) {  
    int n; cin >> n;  
    if(n == 0) break;  
    cout << psum[2 * n] - psum[n] << '\n';  
}
```

# 게임 이론

각 플레이어가 항상 최적의 결정을 내린다고 할 때 ,

누가 이기는지 ?

# 게임 이론

베스킨 라빈스 31 게임은

\*( 번갈아가며 3 이하의 숫자를 말하고 31 을 말하는 사람이 진거 )

1 대 1 로 한다고 할 때 , 시작부터 승자가 정해져 있음

무조건 선공이 이긴다

# 게임 이론

상대	나	
1	2	
	3	
	4	
2	3	...
	4	
	5	
3	4	
	5	
	6	

모든 경우를 전부 계산해 볼 수 있다.

# 게임 이론

또한

단순히 생각해보았을 때  
상대 차례에 27 이상 에서 끝나면 무조건 이김  
27) 28, 29, 30  
28) 29, 30  
29) 30

-> 한 단계 더 가서

상대 차례에 23 이상에서 끝나면 무조건 이김  
(= 내가 26 으로 끝내면 )  
23) 24 25 26  
24) 25 26  
25) 26

# 게임 이론

20004 번 : 베스킨라빈스 31 ( 실버 4)

선공을 뺏긴 상태에서 ,

규칙을 변경해 3 개가 아닌  $n$  개 이하의 수를 부를 수 있다고 할 때

$n$  이 몇일 때 내가 이길 수 있을까



# 게임 이론

함수 정의

$go(now)$  = 해당 차례에  $now$  에서 끝났을 때 , 이길 수 있는지 ? 1 or 0

간단히 말하자면  $go(3)$  이라는 건 내가 마지막에 3 을 말했고 상대 차례로 넘어갔다는 것

$(나\ 1, 2, 3) == (상대\ 1, 나\ 2, 3) == (상대\ 1, 2\ 나\ 3) == go(3)$  모두 같은 상태임

$(나\ 1, 상대\ 2, 나\ 3) =>$

이 상태에서 이길 수 있는가 ?

그렇다면  $go(0)$  이 상대 차례부터 시작하는 것이겠지요

$go(0)$  이 1 이라면 이긴다는거

# 게임 이론

```
if(now == 31) return 0;
```

```
int ret = 1;
```

```
for(int i = 1; i <= 3; i++) {
```

```
    if(now + i > 31) continue;
```

```
    ret = min(ret, 1 - go(now + i)); // 다음으로 진행
```

```
}
```

```
return ret;
```

# 게임 이론

현재 30 이야, 그러면 다음은 31 밖에 못 옴

$ret = 1;$

$ret = \min(ret, 1 - go(31)) = \min(1, 1 - 0) = 1;$

$go(30) \Rightarrow 1$  무조건 이긴다.

현재  $go(29)$ , 그러면 다음은 30, 31 올 수 있음

$ret = 1;$

$ret = \min(ret, 1 - go(30)) = 0;$  // 30 을 고르는 경우 . 위에서  $go(30)$  은 1 이었음

$ret = \min(ret, 1 - go(31)) = 0;$  // 31 을 고르는 경우

$go(29) \Rightarrow 0$  무조건 진다.

여러 가지 경우 중에 지는 경우가 하나라도 있다면 지는 것임 .

이게 바로 각자가 최적의 판단을 한다는 것이다

$\Rightarrow$  내 차례에 29 를 말하고 끝났는데 상대방이 31 을 말하진 않을 것 .

```
if(now == 31) return 0;
```

```
int ret = 1;
```

```
for(int i = 1; i <= 3; i++) {
```

```
    if(now + i > 31) continue;
```

```
    ret = min(ret, 1 - go(now + i));
```

```
}
```

```
go(now = 0);
```

# 게임 이론

아까 말한 문제를 풀려면 3 을 일반화시켜주어야 한다 .

```
if(now == 31) return 0;
int ret = 1;
for(int i = 1; i <= size; i++) {
    if(now + i > 31) continue;
    ret = min(ret, 1 - go(now + i)); // 다음으로 진행
}
return ret;
```

그리고 또 ..

# 게임 이론

겉치는 계산이 너무 많아서 메모이제이션을 해주어야 한다 .  
매 분기마다 size 만큼 갈래를 펼치기 때문에 계산량이 어마어마해짐  
이전에 말했던 것처럼

( 나 1, 2, 3) == ( 상대 1, 나 2, 3) == ( 상대 1, 2 나 3)  
== ( 나 1, 상대 2, 나 3) go(3)  
과 같이 겹침 .

<https://www.acmicpc.net/source/share/cbb5598beacf4637be1ea04d5f22a4ec>

# 게임 이론

쉬운 문제

20004 베스킨라빈스 31

9655 돌 게임

11867 박스 나누기 게임

-> 게임 이론 태그 확인