

Week 16

2024-10-17

종이의 개수

재귀

`go(int x, int y, int k)`

X, y, 에서 시작해서 한 변이 k 인 범위 체크

정답은 `go(0, 0, n)`

`go(x, y, k) =>` 전부 동일하거나 `k == 1` 이라면 정답에 추가하고 `return`

`=>` 전부 동일하지 않다면 `go(x, y + k / 3, k / 3), go(x, y + (k / 3) * 2, k / 3) ..` 의 내부 9 개 범위로 쪼개짐

입력받을때 1 씩 더해서 0, 1, 2 로 값을 바꿔주면 풀기편함

`int ans[3];`

문자열 판별

DP

문자열의 길이들이 짧고 N 도 작아서 그냥 계속 대입해보면 됨

```
int dp[input.size() + 1];
```

dp[i] == 1 라는건 [0, i) 까지의 문자열을 만들 수 있다는 것

dp[0] = 1 로 초기화

오른쪽을 열린 범위로 두는 게 편했음

문자열 판별

dp

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

s	o	f	t	w	a	r	e	c	o	n	t	e	s	t
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

f	t	w	a	r	e
---	---	---	---	---	---

o	n	t	e	s	t
---	---	---	---	---	---

문자열 판별

dp 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

s o f t w a r e c o n t e s t

o f t w a r e

c o n t e s t

문자열 판별

dp 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

s o f t w a r e c o n t e s t

s o f t w a r e

c o n t e s t

문자열 판별

dp 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

s o f t w a r e c o n t e s t

s o f t w a r e

c o n t e s t

문자열 판별

dp 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

s o f t w a r e c o n t e s t

s o f t w a r e

c o n t e s t

문자열 판별

dp 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

s o f t w a r e c o n t e s t

s o f t w a r e

c o n t e s t

문자열 판별

dp 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1

s o f t w a r e c o n t e s t

s o f t w a r e

c o n t e s t

좋은 수열

모든 경우를 만들어가면서 계속 체크를 해주면 됨

만드는 순서는 1, 2, 3 으로 붙여나가면 , 처음으로 나오는 좋은 수열이 정답이 됨

1, 2, 3 으로 만들 수 있는 수의 경우의 수는 3^{80} 이지만 떨어져나가는 게 많아서
실제 가능한 경우의 수는 많지 않음

나머지 합

누적합의 나머지를 구해주고 ,
해당 나머지가 나오는 구간이 몇 개인지 배열에 저장해둬 .

그러면

$$ans = cnt[0] + \sum_{i=0}^{m-1} cnt[i] C_2$$

$$= cnt[0] + \sum_{i=0}^{m-1} \frac{cnt[i] * (cnt[i] - 1)}{2}$$

나머지 합

나머지 배열 대략 이런 모습이라면

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

cnt[0] 은

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

이 부분들이 0 이라는 뜻이고

나머지 합

나머지 배열 대략 이런 모습이라면

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

시그마에 들어가는 부분은

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

나머지연산은
아무곳에나 다
집어넣어도 됨

$cnt[i] C_2$ 두 개를 골라주는것

나머지 합

나머지 배열 대략 이런 모습이라면

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

시그마에 들어가는 부분은

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

1		2		0		1	0	2	1			0	2	
---	--	---	--	---	--	---	---	---	---	--	--	---	---	--

$cnt[i] C_2$

나머지 합

```
cin >> n >> m;
for(int i = 1; i <= n; i++) {
    cin >> a[i];
    a[i] %= m;
    psum[i] = psum[i - 1] + a[i];
    psum[i] %= m;
    cnt[psum[i]] += 1;
}
```

```
ll ans = 0;
for(int i = 0; i < m; i++) {
    ans += (cnt[i] * (cnt[i] - 1)) / 2LL;
}
ans += cnt[0];
```



```
cin >> n >> m;
ll ans = 0;
for(int i = 1; i <= n; i++) {
    cin >> a[i];
    a[i] %= m;
    psum[i] = psum[i - 1] + a[i];
    psum[i] %= m;

    ans += cnt[psum[i]];
    if(psum[i] == 0) ans += 1;

    cnt[psum[i]] += 1;
}
```