4월 모의대회

AIKon 월간 모의대회 해설집

Div. 2	Div. 1	문제	난이도
A		벼락치기	B2
B		창영이와 버스	B1
C		학번	S5
D		재밌는 나머지 연산	S3
E	A	타노스는 요세푸스가 밉다	S2
	B	3으로 나누어 떨어지지 않는 배열	G5
	C	섬 여행	G3
	D	초콜릿 트리 만들기	G2
	E	이진 검색 트리	P5

2A. 벼락치기

- N을 7로 나누었을 때, 평균 영상 개수 (4일에 보는 영상 개수)가 나온다
- 1일에 봐야하는 영상의 개수는 4일보다 3개가 많다
- 즉, 봐야하는 영상의 개수는 $\lceil \frac{N}{7} \rceil + 3$개이다
- 21보다 작거나 같은 N에 대해서는 영상을 안 보는 날이 생기기 시작하기에, 따로 처리를 해주어야 한다
- N이 int를 벗어남을 유의하자

2B. 창영이와 버스

- 2차원 배열로 ‘A[S][E]=환승 요금’ 과 같이 저장한다
- 타야하는 버스의 번호 순서대로 A에 접근해서 요금을 더해주면 된다

2C. 학번

- 학번 중 가장 큰 학번으로 나눈다면 학번을 나눈 나머지는 모두 다를 것이다
- 따라서 1부터 차례대로 나눠보며 나머지가 중복되는 것이 있는지 확인해보면 된다

2D. 재밌는 나머지 연산

- $N = mQ + R$로 나타낼 수 있다 (Q는 몫)
- $mQ = N - R$이어야 하므로, 정수 m은 $N - R$의 소인수이다
- 나누는 수가 나머지보다 커야 한다 ($m > R$)
- 위 조건을 만족하는 m의 합을 출력하면 된다

2E/1A. 타노스는 요세푸스가 밉다

- 큐로 문제의 내용을 그대로 구현해주면 된다!
- 첫 번째 청설모의 오른쪽 청설모가 첫 번째 청설모가 되는 과정은 큐에서 `front`를 `pop`하고 다시 `push`해주는 과정과 같다

1B. 3으로 나누어 떨어지지 않는 배열

- 3으로 나누어 떨어진다는 것은 3으로 나누었을 때의 나머지가 0이라는 것이다
- 모듈러 연산의 성질에 의해 $(a + b) \bmod c = (a \bmod c + b \bmod c) \bmod c$가 성립
- 배열의 원소들을 3으로 나눈 나머지가 0, 1, 2인 것으로 나눈다
- 나머지의 합이 3으로 나누어 떨어지는 (0-0, 1-2, 2-1)과 같이 배치해서는 안 된다

1C. 섬 여행

- A에서 어떤 정점 x를 특정 경로로 이동했을 때, 거쳐간 다리 수를 k라고 하자
- 정점 x와 인접한 정점 y를 거치고 다시 x로 돌아오는 상황을 상상해보자
- 그러면 거쳐간 다리 수는 $k, k+2, k+4, ... = k+2t$꼴이고 이 값이 K여야 한다
- $k+2t=K$이므로 $(K-k)\%2=0$ → $K\%2=k\%2$ 이다
- D(x, r)을 A에서 어떤 정점 x까지 최소로 거쳐간 다리 수 (최소로 거쳐간 다리 수 % 2 = r)이라고 하면, D(x, $K\%2$) $\leq K$를 만족하는 정점 x들을 찾으면 된다
- D는 A에서 BFS를 이용해 값을 채울 수 있다

1D. 초콜릿 트리 만들기

- 정수 $k(0 \le k < N)$에 대해 $M = k + M - k$ 또는 $M + N = k + M + N - k$
- 따라서 부모가 M인 초콜릿을 얻기 위해 올 수 있는 자식은

$k \le M$이면 $(k, M - k)$

$k > m$이면 $(k, M + N - k)$이다

- 부모가 M인 초콜릿을 최소 개수로 얻으려면 자식 초콜릿도 최소로 써야한다
- `optimal substruct`가 성립하므로 DP를 이용해 풀 수 있다

1D. 초콜릿 트리 만들기

- DP(h, m)을 깊이가 H - h인 노드에서 숫자가 m인 초콜릿을 얻기 위해 최소로 빌려야 하는 초콜릿 개수로 정의하자
- DP(0, m)은 이미 가지고 있는 초콜릿이면 0, 빌려야 하는 초콜릿이면 1로 초기화하고 나머지는 매우 큰 수로 초기화한다
- bias: m이 이미 가지고 있는 초콜릿이면 0, 빌려야 하는 초콜릿이면 1
- DP(h, m) = min(DP(h, m), DP(h-1, k) + DP(h-1, M-k) + bias) ($k \leq M$)
- DP(h, m) = min(DP(h, m), DP(h-1, k) + DP(h-1, M+N-k) + bias) ($k > M$)

1E. 이진 검색 트리

- 이진 검색 트리에 존재하는 데이터의 집합 S에 대해 추가로 데이터 x를 삽입한다면, S에서 x보다 작은 원소 중 최대인 원소 s_1의 오른쪽 자식 또는 S에서 x보다 큰 원소 중 최소인 원소 s_2의 왼쪽 자식에 올 것이다
- 따라서, x의 높이는 $s_1 + 1$ 또는 $s_2 + 1$이다
- 이진 검색 트리의 성질에 의해 s_1과 s_2는 삽입된 순서에 따라 s_1과 s_2는 직접적인 부모 - 자식 관계이기 때문에, 두 상황 중 하나의 상황만 올 것이다
- 그렇기에 s_1의 오른쪽 자식이 이미 존재한다면, x는 s_2의 왼쪽 자식이다
- s_2의 왼쪽 자식이 이미 존재한다면, x는 s_1의 오른쪽 자식이다

1E. 이진 검색 트리

- S에서 x보다 작은 원소 중 최대인 원소 s_1을 구하기 위해 최댓값 세그먼트 트리를 활용한다
- S에서 x보다 큰 원소 중 최소인 원소 s_2를 구하기 위해 최솟값 세그먼트 트리를 활용한다
- 각 세그먼트 트리에서는 이진 검색 트리에 데이터 x를 삽입했다면 인덱스 x의 value를 x로 업데이트 한다
- 그리고 각 데이터마다 트리의 높이를 저장하는 길이 N인 배열과 각 데이터마다 왼쪽 자식과 오른쪽 자식이 존재하는지 알려주는 길이 $N * 2$짜리 배열이 필요하다