제1회 대전과학고등학교 프로그래밍 경진대회 DSHStack 풀이

Official Solution

by

제1회 DSHStack 출제진 및 검수진

2024년 12월 16일



출제진

- ✓ koi312500
- ✓ leessjn
- ✓ ohwphil
- ✓ potterlovejh
- ✓ wider93



검수진

- ✓ dongwook7
- ✓ hyperbolic
- ✓ iktk
- ✓ sjhi00
- ✓ woohyun_jng

문제		의도한 난이도	출제자
A	왜 맘대로 예약하냐고	**Easy**	koi312500
B	그래프와 그래프	**Easy**	potterlovejh
C	자리 신청	**Easy**	koi312500
D	잘못된 LIS 알고리즘	**Easy**	ohwphil
E	극한직업 - 영양사 선생님	**Medium**	ohwphil
F	오코노미야키 만들기	**Medium**	ohwphil
G	배달하기	**Medium**	koi312500, ohwphil
H	삼각형의 부분합	**Medium**	wider93
I	아침 루틴과 아침 점호	**Medium**	leessjn
J	점화식과 쿼리	**Hard**	wider93

A. 왜 맘대로 예약하냐고

implementation
출제진 의도 – **Easy**
출제자 : koi312500

교내 대회

- ✓ 제출 54번, 정답 34명 (정답률 62.963%)
- ✓ 처음 푼 사람: **K**, 2분

Open Contest

- ✓ 제출 158번, 정답 136명 (정답률 87.342%)
- ✓ 처음 푼 사람: **dabbler1**, 0분

A. 왜 맘대로 예약하냐고

- N 명의 조원 모두가 건우가 예약한 교시 X 에 참석 가능해야 합니다.
- 즉, 각 조원마다 X 교시에 참석 가능한지 확인하면 됩니다.
- 따라서 입력받은 조원마다 참석이 가능한지를 체크해, 모든 조원이 참석 가능하면 해당 시간에 신청이 가능합니다.

B. 그래프와 그래프

graph theory, bruteforcing
출제진 의도 – **Easy**
출제자 : potterlovejh

교내 대회

- ✓ 제출 62번, 정답 32명 (정답률 51.613%)
- ✓ 처음 푼 사람: **K**, 6분

Open Contest

- ✓ 제출 171번, 정답 110명 (정답률 67.836%)
- ✓ 처음 푼 사람: **dabbler1**, 3분

B. 그래프와 그래프



- ✓ x와 y가 1부터 10까지의 자연수일 때 등식을 만족하는 (x, y) 쌍을 모두 찾으면 됩니다.
- ✓ x와 y에 1부터 10까지의 자연수를 넣으며 등식이 성립하는지 확인하면 됩니다.

C. 자리 신청

`implementation, simulation, sorting`
출제진 의도 – **Easy**
출제자 : `koi312500`

교내 대회

- ✓ 제출 91번, 정답 25명 (정답률 27.473%)
- ✓ 처음 푼 사람: **K**, 10분

Open Contest

- ✓ 제출 94번, 정답 64명 (정답률 70.213%)
- ✓ 처음 푼 사람: **dabbler1**, 5분

C. 자리 신청

✓ 각 로그는 3가지로 나누어 생각할 수 있습니다.

1. 신청한 좌석이 **신청된 좌석**일 때
   - ▶ 이미 누군가에 의해 **신청된 좌석**이면 해당 신청은 무효화됩니다.
2. 신청한 좌석이 **신청되지 않은 좌석**이고, 신청자가 **기존에** 어떤 자리도 **배정되지 않았을 때**
   - ▶ 신청한 좌석에 배정합니다.
   - ▶ 학생이 어디에 앉았는지 체크합니다.
3. 신청한 좌석이 **신청되지 않은 좌석**이고, 신청자가 **기존에** 이미 다른 자리에 **배정된 경우**
   - ▶ 신청한 좌석에 배정합니다.
   - ▶ 기존에 앉았던 좌석을 **신청되지 않은 좌석**으로 만듭니다.
   - ▶ 학생이 어디에 앉았는지 체크합니다.

C. 자리 신청

- ✓ 신청 로그를 시간 순서대로 정렬한 뒤 각 신청을 차례대로 처리합니다.
- ✓ 위의 조건에 따라 각각의 경우에 맞게 신청을 처리합니다.
- ✓ 이 때, 좌석과 학생의 배정 정보는 배열을 사용해 처리합니다.
- ✓ 모든 로그 처리가 완료된 후 학생들의 최종 자리 배치를 학번 순서대로 출력합니다.

D. 잘못된 LIS 알고리즘

ad hoc, constructive
출제진 의도 – **Easy**
출제자 : ohwphil

교내 대회

- ✓ 제출 81번, 정답 23명 (정답률 28.395%)
- ✓ 처음 푼 사람: **아스제**, 14분

Open Contest

- ✓ 제출 66번, 정답 40명 (정답률 63.636%)
- ✓ 처음 푼 사람: **dabbler1**, 11분

D. 잘못된 LIS 알고리즘

- ✓ 우선 해를 구성하는 것이 불가능한 경우부터 살펴봅시다.
- ✓ 만약 A의 가장 긴 증가하는 부분 수열(이하 LIS)의 길이가 M이 N과 같다면, 해를 구성하는 것이 불가능합니다. 그 이유를 알아봅시다.
- ✓ $M = N$이라는 조건 때문에 $A = [1, 2, \cdots, N]$으로 고정되는데, 여기에 정현이의 알고리즘을 적용하면 반환하는 부분 수열의 길이가 무조건 N이 됩니다.
- ✓ 제약조건에 의해 $K < M = N$인 입력만 주어지는데, 정현이의 알고리즘이 반환하는 부분 수열의 길이가 정확히 N일 수밖에 없기 때문에 $M = N$인 경우는 조건을 만족하는 해를 구성할 수 없습니다.
- ✓ 그런데, 놀랍게도 이 경우 외에는 해가 항상 존재합니다!

D. 잘못된 LIS 알고리즘

- ✓ 해를 구성하는 방법을 생각해보기 전에 유사코드의 로직을 관찰해 봅시다.
- ✓ $A_1, A_2, \cdots, A_i$ 중에서 최댓값을 m 이라고 합니다.
- ✓ $i+1$ 번째 원소 이후로 m 보다 작은 원소가 나오면 이 원소는 B 에 추가되지 않고 무시된다는 것을 알 수 있습니다.
- ✓ 따라서 정현이의 알고리즘이 반환하는 부분 수열 B 의 길이가 K 가 되게 하기 위해서 적당히 큰 원소를 잘 활용해야 할 것 같다는 생각을 할 수 있습니다.

D. 잘못된 LIS 알고리즘



- ✓ 여기서는 한 가지 해법만 제시해 보겠습니다.
- ✓ N 이라는 원소를 활용하여 정현이의 알고리즘이 길이 K 인 수열을 반환하도록 해 보겠습니다.
- ✓ A 의 맨 앞 $K-1$ 개 원소를 1 부터 $K-1$ 까지의 정수로 두고, K 번째 원소를 N 으로 두면 이 뒤에 무엇이 오든지 상관없이 정현이의 알고리즘은 길이 K 인 수열 $[1, 2, \cdots, K-1, N]$ 을 반환합니다.
- ✓ 이제 나머지 원소들을 잘 배열하면 되는데, 느낌상 LIS가 앞서 정해둔 맨 앞의 $K-1$ 개 원소와 맞물려서 정해졌으면 좋을 것 같습니다.
- ✓ $K+1$ 번째 원소부터 $N-M+K$ 번째 원소까지를 $N-1, N-2, \cdots, M+1$ 로 두고, $N-M+K+1$ 번째 원소부터 N 번째 원소까지를 $K, K+1, \cdots, M$ 으로 둡니다.

D. 잘못된 LIS 알고리즘



- ✓ 그리하면 $A = [1, 2, \cdots, K-1, N, N-1, N-2, \cdots, M+1, K, K+1, \cdots, M]$ 이 되고, 이 수열은 조건을 만족합니다.
- ✓ 해 구성을 $\mathcal{O}(N)$ 에 할 수 있습니다.

E. 극한직업 - 영양사 선생님

`greedy, sorting`
출제진 의도 – **Medium**
출제자 : `ohwphil`

교내 대회

- ✓ 제출 88번, 정답 18명 (정답률 21.591%)
- ✓ 처음 푼 사람: **K**, 20분

Open Contest

- ✓ 제출 73번, 정답 43명 (정답률 60.274%)
- ✓ 처음 푼 사람: **fermion5**, 3분

E. 극한직업 - 영양사 선생님



- 이 문제에는 크게 2가지 방향의 풀이가 있습니다.
- 풀이 1. 시점 기준으로 생각하기
- 가능한 최대 혼잡도를 M 이라고 가정해 봅시다. 그리고 $\{b_i\}$ 라는 수열을 재배열한 수열 $\{c_i\}$ 를 생각해 봅시다.
- 학생들을 배열한 결과 줄의 앞에서부터 i 번째 학생이 c_i 분만큼 밥을 먹을 때 혼잡도가 M 이 되었다고 합시다.
- 그리고 급식 시간이 시작된 지 T 분 후에 M 명의 학생들이 앉아 있다고 가정해 봅시다.
- T 분 시점에 앉아 있는 학생들이 각각 $T - d_j (1 \leq j \leq M; d_j \geq 0)$ 분에 배식을 받았다면, 이 학생들은 $d_j + 1$ 분 이상 동안 밥을 먹어야 합니다.

E. 극한직업 - 영양사 선생님



- ✓ 일반성을 잃지 않고, d_j 들은 오름차순으로 정렬되어 있다고 가정하겠습니다.
- ✓ 그렇다면 $d_j \geq j-1$ 이라는 사실을 알 수 있습니다.
- ✓ 따라서 만약 $T-d_j$ 분에 배식을 받은 학생이 T 분에 앉아 있다면, 만약 그 학생이 $T-j+1$ 분에 배식을 받는다고 해도 T 분에 앉아 있습니다. 학생이 앉아 있는 구간이 전반적으로 뒤로 밀려났기 때문입니다.
- ✓ 그러므로 $T, T-1, \cdots, T-M+1$ 분에 배식받은 학생이 T 분에 앉아 있는 해도 존재한다고 할 수 있습니다.
- ✓ T 분에 배식받은 학생은 1 분 이상, $T-1$ 분에 배식받은 학생은 2 분 이상, ... 식사를 해야 합니다.

E. 극한직업 - 영양사 선생님

- ✓ 그렇다면 반대로 해를 구성하는 입장에서는, 학생들이 식사하는 시간을 오름차순 정렬한 후 1분 이상 먹는 학생을 한 명 고르고, 2분 이상 먹는 학생을 한 명 고르고, ... 하면서 몇 명의 학생을 고를 수 있는지 세면 그것이 답이 됩니다.
- ✓ 이 순서대로 고르면서, 최대한 밥 먹는 시간이 작은 학생부터 고르는 것이 최적입니다. (greedy한 전략)

E. 극한직업 - 영양사 선생님



- 풀이 2. 학생들을 내림차순 정렬 후 최대 혼잡도 계산
- 결론부터 이야기하자면, 학생들이 밥을 먹는 시간을 기준으로 내림차순 정렬한 후 최대 혼잡도를 계산해도 됩니다.
- 일단 풀이 1에서와 같이 시점 기준으로 생각해 봅시다.
- 만약 $i < j$ 이고, $T - i$ 분에 배식을 받는 학생과 $T - j$ 분에 배식을 받는 학생이 모두 T 분에 급식실에 앉아 있다고 합시다.
- $T - i$ 분에 배식을 받는 학생은 k 분 동안 식사하고, $T - j$ 분에 배식을 받는 학생은 l 분 동안 식사하는데, $k > l$ 이라고 해 봅시다.
- 만약 이 학생들이 배식 받는 순서를 바꿔서, l 분 동안 식사하는 학생이 $T - i$ 분에 대신 배식을 받고, k 분 동안 식사하는 학생이 $T - j$ 분에 배식을 받는다고 합시다.

E. 극한직업 - 영양사 선생님



- ✓ 이 두 학생이 배식받는 순서를 바꾸어도 두 학생은 모두 T 분에 앉아 있습니다! (증명 생략)
- ✓ 따라서 한 시점에 대해서 생각할 때 배식을 받는 시간 기준으로 내림차순 정렬을 하는 것이 맞다는 사실을 알 수 있습니다.
- ✓ 그렇다면 전체 학생들을 내림차순 정렬한 후 최대 혼잡도를 구하는 방법은 최적해일까요? 이것 역시 답이 "예"입니다.
- ✓ 증명을 생각해 보면, 어떤 시점 T 에 M 명의 학생이 앉아 있다고 합시다. 풀이 1의 결과를 인용하면, $T-M+1$ 분, $T-M+2$ 분, ..., T 분에 학생이 배식을 받았다고 가정할 수도 있습니다.
- ✓ 이때 학생들이 밥을 먹는 시간이 각각 $x_1, x_2, \cdots, x_M$ 분이라고 하고, 전체 학생들을 내림차순 정렬했을 때 처음 M 명의 학생들이 밥을 먹는 시간을 $y_1, y_2, \cdots, y_M$ 분이라고 합시다.

E. 극한직업 - 영양사 선생님



- ✓ 그렇다면 $x_1, x_2, \cdots, x_M$ 분 동안 밥을 먹는 학생들 대신 $y_1, y_2, \cdots, y_M$ 분 동안 밥을 먹는 학생들으로 대체해도 M 명의 학생들이 T 분에 자리에 앉아 있을 것입니다!
- ✓ 따라서 학생들이 밥을 먹는 시간을 내림차순 정렬한 후 그에 대응되는 구간들을 구한 후, 이 구간들 중에서 최대한 많이 겹칠 때 몇 개의 구간이 겹치는지 세 주면 자연스럽게 최댓값이 나오게 됩니다.
- ✓ 이 문제는 BOJ 11000번 (강의실 배정) 문제의 방법으로 $\mathcal{O}(N \log N)$ 시간에 해결하거나, 구간의 시작 시점의 공차가 1이라는 점을 활용하여 다양한 방법으로 $\mathcal{O}(N)$ 시간에 해결할 수 있습니다.

E. 극한직업 - 영양사 선생님



- ✓ 두 전략 모두 정렬하는 데에 $\mathcal{O}(N \log N)$ 시간이 들어가며, 정렬 후 답을 구하는 과정에는 $\mathcal{O}(N)$ 시간이 들어갑니다.
- ✓ cf. 이 두 greedy 전략의 증명은 최적해를 하나 두고, 이 최적해를 변형시켜 greedy한 방법으로 얻은 해를 이끌어냄으로써 greedy 전략의 해가 최적해라는 것을 증명하는 방식으로 이루어졌습니다. 이러한 증명 방식을 exchange argument라고 부릅니다.

F. 오코노미야키 만들기

ad hoc, implementation, casework
출제진 의도 – **Medium**
출제자 : ohwphil

교내 대회

- ✓ 제출 69번, 정답 2명 (정답률 2.899%)
- ✓ 처음 푼 사람: ., 90분

Open Contest

- ✓ 제출 88번, 정답 18명 (정답률 20.455%)
- ✓ 처음 푼 사람: **xiaowuc1**, 33분

F. 오코노미야키 만들기

- 우선 문제 해설을 시작하기 전에, 번호가 작아지는 쪽을 **왼쪽**, 커지는 쪽을 **오른쪽**이라고 지칭하겠습니다.
- 고기가 상대적으로 왼쪽에 있는 오코노미야키 반죽에 붙은 뒤 S번 칸으로 이동할 수도 있고, 오른쪽에 있는 오코노미야키 반죽에 붙은 뒤 S번 칸으로 이동할 수도 있습니다.
- 두 반죽 모두에 대하여 고기가 붙고 S번 칸으로 이동하는 것이 불가능하다면 -1을 출력하면 됩니다.
- 두 반죽 중 하나에 대해서만 가능하다면 그 반죽에 대해서 필요한 최소 횟수를 출력하면 됩니다.
- 두 반죽 모두에 대해서 가능하다면 두 반죽에 대해서 필요한 최소 횟수를 구한 뒤 최솟값을 출력하면 됩니다.

F. 오코노미야키 만들기



- 그렇다면 일반성을 잃지 않고, 왼쪽 반죽에 고기가 붙고 S 번 칸에 있도록 한다고 가정합니다.
- 우선 왼쪽 반죽이 처음에 p_1 번 칸에 있고 덜 익은 면이 위에 올라와 있습니다.
- 왼쪽 반죽이 X 번 칸에 있을 때 p_1 과 X 를 각각 2 로 나눈 나머지가 동일하면 반죽의 덜 익은 면이 위로 올라오고, 동일하지 않다면 덜 익은 면이 바닥으로 갑니다.
- 따라서 M 과 p_1 을 2 로 나눈 나머지가 동일하다면 왼쪽 반죽에 고기를 붙여 목표를 달성하는 것은 불가능합니다.
- 또한 반죽이 있는 칸으로 뒤집는 것이 불가능하기 때문에, 왼쪽 반죽과 오른쪽 반죽의 위치관계가 서로 바뀔 수 없습니다.
- 따라서 M 과 S 중에 단 하나라도 N 과 같다면 왼쪽 반죽에 고기를 붙여 목표를 달성하는 것은 불가능합니다.

F. 오코노미야키 만들기

- 이제는 왼쪽 반죽을 M 번 칸으로 이동시키고, 그런 후 S 번 칸으로 이동시키는 경우를 생각해 봅시다.
- 만약 오른쪽 반죽의 위치 p_2 가 M 보다 왼쪽에 있다면, 왼쪽 반죽이 M 번 칸으로 가기 위해서 오른쪽 반죽이 비켜 주어야 합니다. 따라서 오른쪽 반죽을 $M+1$ 번 칸으로 옮겨야 합니다.
- 하지만 그 과정에서 오른쪽 반죽이 고기와 붙게 된다면, 왼쪽 반죽에 고기를 붙여 목표를 달성하는 것은 불가능합니다.
- 그런 다음에, 마찬가지로 오른쪽 반죽이 S 번 칸보다 왼쪽에 있거나 S 번 칸 위에 있다면, 오른쪽 반죽이 $S+1$ 번 칸으로 이동하여 비켜 줘야 합니다.
- 위와 같이 이동하는 것이 최적의 방법입니다.
- 오른쪽 반죽에 고기를 붙이는 경우도 대칭적으로 구해 주시면 됩니다.

F. 오코노미야키 만들기

- 풀이 및 주의할 점을 요약하자면 다음과 같습니다.
- 고기가 왼쪽 반죽에 붙는 경우, 오른쪽 반죽에 붙는 경우로 나누어서 해결합니다.
- 1칸, 3칸, 5칸, ... 등 처음 위치가 고기로부터 홀수 칸 떨어진 반죽에만 덜 익은 면에 고기를 붙일 수 있습니다.
- 두 반죽의 위치가 바뀔 수 없기 때문에 왼쪽 반죽은 N번 칸에, 오른쪽 반죽은 1에 절대로 위치할 수 없습니다.
- 다른 반죽 하나가 비켜 주어야 하는 경우, 그 과정에서 고기가 그 다른 반죽에 붙게 되는 경우에 유의합니다.

G. 배달하기

graph theory, bfs
출제진 의도 – **Medium**
출제자 : koi312500, ohwphil

교내 대회

- ✓ 제출 27번, 정답 1명 (정답률 3.704%)
- ✓ 처음 푼 사람: **K**, 63분

Open Contest

- ✓ 제출 43번, 정답 7명 (정답률 16.279%)
- ✓ 처음 푼 사람: **dkim110807**, 67분

G. 배달하기

- 이 문제는 다익스트라 알고리즘을 활용한 풀이와 BFS를 활용한 풀이가 있습니다.
- 다익스트라를 활용한 풀이는 떠올리기는 상대적으로 쉬우나 구현하기가 상당히 까다로우며, BFS를 활용한 풀이는 기존의 BFS에서 떠올려야 할 추가적인 아이디어가 있지만 구현은 비교적 간단하다는 특징이 있습니다.
- 다익스트라 알고리즘을 활용한 풀이부터 알아보겠습니다.

G. 배달하기

- 풀이 1. 다익스트라
- 우선순위 큐에 귀납적으로 감시당하고 있지 않은 교차점만 들어가게끔 해서 관리합니다.
- 한 교차점에서 다른 교차점으로 이동할 때 이동할 수 있는 가장 빠른 시간을 구하여야 하는데, 구하는 과정에서 선형으로 각 시간이 가능한지 확인하면 안 되고, 이분 탐색 등을 사용하여야 합니다.
- 이때, 이동하려는 교차점이 감시당하고 있을 때의 처리에 특히 주의하셔야 합니다.
- 시간복잡도는 $\mathcal{O}(M \log N \log K)$ 입니다.

G. 배달하기

- 풀이 2. BFS
- 이 알고리즘은 일반적인 BFS를 변형하여 다음과 같은 차이를 둬야 합니다.
- 우선 각 교차점까지 도달하는 데에 걸리는 시간을 담는 배열을 다익스트라 알고리즘에서와 마찬가지로 둬야 합니다.
- 현재 시간이 T 만큼 지났을 때, $T+1$ 분에 이동하려는 목표 정점이 감시당하고 있어도 큐에 push합니다.
- 현재 큐에서 꺼낸 정점이 감시당하고 있다면, 걸리는 시간을 1 증가시킨 후 다시 큐에 담습니다.
- 여기서 유의해야 할 점은, 한 정점만 영원히 감시당하는 경우 무한 루프에 빠질 수 있기 때문에 계속 감시당하는 그 정점은 애초에 없는 정점처럼 취급해야 합니다.
- 시간복잡도는 $\mathcal{O}(K+N+M)$ 임을 보일 수 있습니다.

H. 삼각형의 부분합



`prefix sum`
출제진 의도 – **Medium**
출제자 : `wider93`

교내 대회

- ✓ 제출 16번, 정답 2명 (정답률 12.500%)
- ✓ 처음 푼 사람: **soccer555sm**, 99분

Open Contest

- ✓ 제출 34번, 정답 9명 (정답률 26.471%)
- ✓ 처음 푼 사람: **dabbler1**, 41분

H. 삼각형의 부분합



- 편의상 a_{ij} 를 $a_{i,j}$ 로 표기하겠습니다.
- $1 \le j \le i \le N$ 일 때, b_{ij}, p_{ij}, q_{ij} 를 다음과 같이 정의하겠습니다.
  - $b_{i,j} = \sum_{k=1}^{j} a_{i,j}, \quad b_{i,0} = 0$
  - $p_{i,j} = \sum_{k=j}^{i} b_{k,j}, \quad p_{j-1,j} = 0$
  - $q_{i,j} = \sum_{k=0}^{j-1} b_{i-k,j-k}, \quad q_{0,0} = q_{i,0} = 0$
- 이때, $b_{i,j} = b_{i,j-1} + a_{i,j}, \quad p_{i,j} = p_{i-1,j} + b_{i,j}, \quad q_{i,j} = q_{i-1,j-1} + b_{i,j}$ 입니다.

H. 삼각형의 부분합



- 이때, x, y, z에 의해서 결정되는 부분 정삼각형($0 \le i \le z-1$을 만족하는 모든 정수 i에 대해, 위에서 $x+i$번째 층의 구슬 중 왼쪽에서 y번째부터 $y+i$번째 구슬까지를 포함하는 정삼각형)에 있는 모든 구슬에 적힌 수의 합은 다음과 같이 쿼리 한 개당 $\mathcal{O}(1)$에 구할 수 있습니다.

$$\begin{aligned}
&\sum_{i=0}^{z-1}\left(\sum_{j=0}^{i} a_{x+i,y+j}\right) \\
&= \sum_{i=0}^{z-1}\left(b_{x+i,y+i} - b_{x+i,y-1}\right) \\
&= \left(q_{x+z-1,y+z-1} - q_{x-1,y-1}\right) - \left(p_{x+z-1,y-1} - p_{x-1,y-1}\right)
\end{aligned}$$

- 전체 시간 복잡도는 $\mathcal{O}\left(N^2 + Q\right)$ 입니다.

I. 아침 루틴과 아침 점호

DP, data structures
출제진 의도 – **Medium**
출제자 : leessjn

교내 대회

- ✓ 제출 16번, 정답 1명 (정답률 6.250%)
- ✓ 처음 푼 사람: **K**, 114분

Open Contest

- ✓ 제출 19번, 정답 9명 (정답률 47.368%)
- ✓ 처음 푼 사람: **dbgusdn012**, 50분

- ✓ 어떤 가치를 가진 아이템들을 최적으로 모으려고 한단 점에서 배낭 문제를 떠올릴 수 있습니다.
- ✓ 우선, 단계가 낮은 행동은 언제나 단계가 높은 행동보다 선행되어야 하므로 단계를 기준으로 행동들을 정렬해줍시다.



- ✓ 이 때 평범한 배낭 문제의 풀이처럼 DP 배열을 갱신해줍시다.
- ✓ $DP[i][j]$: j 번째 단계, 또는 그 이전까지 끝났을 때 i 의 시간을 썼을 때 느낄 수 있는 최대 만족감입니다.
- ✓ j 번째 단계인 k 번째 행동에 대해 해당 행동이 주는 만족감을 $M[k]$, 해당 행동이 소비하는 시간을 $t[k]$ 라고 하면
  $DP[i][j] = \max(DP[i][j], \max(DP[i-t[k]][j-1], DP[i-t[k]][j]) + M[k])$ 입니다.
- ✓ $i - t[k]$ 가 0 이상의 정수인지 여부를 고려해야 합니다.
- ✓ 또한 $DP[i-t[k]][j-1]$ 와 $DP[i-t[k]][j]$ 의 실제 방문 여부에 유의해야 합니다.
- ✓ 시간복잡도는 $\mathcal{O}(NT)$ 입니다.

J. 점화식과 쿼리



`math`, `number theory`, `exponentiation by squaring`
출제진 의도 – **Hard**
출제자 : `wider93`

교내 대회

- ✓ 제출 5번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람: —, —분

Open Contest

- ✓ 제출 17번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람: —, —분

- ✓ 주어진 점화식은 선형점화식이 아니지만, 선형점화식에 관한 문제로 쉽게 바꿀 수 있습니다. 이 부분을 관찰해야(혹은 알고 있어야) 이 문제를 풀 수 있습니다.
- ✓ 두 가지 방법을 소개하겠습니다. 편의상 소수 $p = 100003$ 으로 둡시다.

J. 점화식과 쿼리



- ✓ 조금 더 쉽게 접근할 수 있는 방법입니다만, 여전히 최소한의 사전 지식은 필요합니다. 행렬 곱셈을 이용하여 선형점화식을 구하는 방법을 알면 풀이에 접근할 수 있습니다.
- ✓ $y_n := x_{n+p} - x_n$ 를 정의하면, y_n 은 점화식 $y_n \equiv a y_{n-1} + b y_{n-2} \mod p$ 를 만족합니다.
- ✓ 이제 구하는 값은 $x_n \equiv x_{n\%p} + (y_{n\%p} + \cdots + y_{n-p})$ 가 됩니다.
- ✓ 행렬 $A \equiv \begin{pmatrix} a & b \\ 1 & 0 \end{pmatrix}$ 로 두었을 때 $\begin{pmatrix} y_{n+1} \\ y_n \end{pmatrix} \equiv A \begin{pmatrix} y_n \\ y_{n-1} \end{pmatrix} \equiv \cdots = A^n \begin{pmatrix} y_1 \\ y_0 \end{pmatrix}$ 을 만족합니다.
- ✓ 따라서 $\begin{pmatrix} x_{n+1} \\ x_n \end{pmatrix} \equiv \begin{pmatrix} x_{n\%p+1} \\ x_{n\%p} \end{pmatrix} + (I + A + \cdots + A^{(n/p-1)p}) \begin{pmatrix} y_{n\%p+1} \\ y_{n\%p} \end{pmatrix}$ 또한 성립합니다.
- ✓ 따라서 $x_2, \cdots, x_p$ 까지를 미리 구해 놓고, 분할정복을 이용한 거듭제곱을 통해 A^p 와 $I + A^p + \cdots + A^{(n/p-1)p}$ 를 구하면 $\mathcal{O}(p \log k + Q \log(n/p))$ 에 문제를 해결할 수 있습니다.



- ✓ 다른 풀이는 특성다항식을 이용한 풀이입니다.
- ✓ 해당 개념을 모른다면, 다음의 정의를 사용합시다: 수열 a_n 이 모든 정수 $n \geq 0$ 에 대해 점화식 $a_{n+m} + b_{m-1}a_{n+m-1} + \cdots + b_0 a_n = 0$ 을 만족할 때, 수열 a_n 이 특성다항식 $x^m + b_{m-1}x^{m-1} + \cdots + b_0$ 를 갖는다고 말합니다.
- ✓ 수열 a_n 의 특성다항식 $p(x)$ 에 대하여, p 로 나누어지는 다른 다항식 q 도 a_n 의 특성다항식이 됨을 보일 수 있습니다.
- ✓ k 차 다항식 p 에 대해, 수열 $x_n = ax_{n-1} + bx_{n-2} + p(n)$ 이 특성다항식 $(x^2 - ax - b)(x-1)^{k+1}$ 을 가진다는 것을 귀납법으로 보일 수 있습니다.
- ✓ 위 다항식에 의한 점화식을 바로 사용하기는 곤란합니다. $(k+3) \times (k+3)$ 행렬의 거듭제곱을 이용하여 일반항을 구하기 위해서는 $\mathcal{O}(k^3)$ 의 행렬 곱셈이 지나치게 비싼 연산입니다.

J. 점화식과 쿼리

- ✓ 정수 $0 < m < p$에 대하여, 이항계수 $\binom{p}{m}$는 항상 p의 배수임을 떠올립시다.
- ✓ 그러면 정수계수 다항식 $P(x)$에 대해, $P(x)^p \equiv P(x^p) \mod p$임을 보일 수 있습니다.
- ✓ 따라서 $3p$차 다항식 $(x^{2p} - ax^p - b)(x^p - 1) \equiv (x^2 - ax - b)^p (x-1)^p$가 법 p에 대해 $(x^2 - ax - b)(x-1)^{k+1}$로 나누어떨어지는 다항식이 됩니다.
- ✓ 따라서 수열 x_n은 법 p에 대해 점화식 $x_{n+3p} - (a+1)x_{n+2p} - (b-a)x_{n+p} + b = 0$을 만족합니다. 즉 $y_m := x_{n_0+mp}$꼴의 수열을 정의하면 y_n은 3차 선형점화식을 만족하여, $x_0, \cdots, x_{3p-1}$을 계산해 두고, 쿼리마다 $y_{n/p}$을 구하는 것으로 문제를 해결할 수 있습니다.
- ✓ 시간복잡도는 마찬가지로 $\mathcal{O}(p \log k + Q \log(n/p))$가 됩니다.
- ✓ 도전: $x_0, \cdots, x_{3p}$를 법p로 $\mathcal{O}(p)$에도 구할 수 있을까요?