SPC 2024

2024 제20회

서강대학교 프로그래밍 대회

Sogang Programming Contest 2024

에디토리얼

주최 서강대학교 Sogang University 서강대학교 소프트웨어중심대학사업단 주관 Sogang ICPC Team

후원 과학기술정보통신부 IITP 정보통신기획평가원

Samsung Software Membership HYUNDAI MOBIS 한빛미디어 SOLVED.AC STARTLINK

Master

문제		의도한 난이도	출제자
A	SPC에 가는 길	**Easy**	이승형 tmdgud0617
B	문자열 줄이기	**Easy**	박준영 sparkjy18
C	두 스택	**Easy**	송근수 hulahula3247
D	$A_i \times A_j$	**Easy**	송근수 hulahula3247
E	gcd와 최단경로	**Medium**	강효규 djs100201
F	서강 피자	**Medium**	조원빈 wbcho0504
G	근수의 미로게임	**Hard**	송근수 hulahula3247
H	물통	**Hard**	강효규 djs100201

A. SPC에 가는 길

ad_hoc
출제진 의도 – **Easy**

- 제출 175번, 정답 42명 (정답률 24.000%)
- 처음 푼 사람: **박성준**, 3분
- 출제자: 이승형 `tmdgud0617`

- 우선 근수가 존재하지 않을 때의 답에 대해 생각해봅시다.
- 승형이의 집과 SPC 대회장의 x 좌표 혹은 y 좌표가 같다면 방향 전환 없이 도달할 수 있습니다.
- 그렇지 않다면, 반드시 1 회의 방향 전환이 필요합니다.
- 이 경우에는 2 회 이상의 방향 전환은 필요하지 않습니다.

- 이제, 근수의 위치에 따른 답의 변화를 살펴봅시다.
- 승형이의 집과 SPC 대회장의 x좌표 혹은 y좌표가 같을 때, 근수가 그 사이에 위치하지 않는다면 여전히 방향 전환 없이 도달할 수 있습니다.
- 근수가 승형이의 집과 SPC 대회장 사이에 위치한다면, 2회의 방향 전환으로 도달할 수 있습니다.
- 기존에 1회의 방향 전환으로 도달할 수 있었던 SPC 대회장은 근수의 위치와 관계없이 여전히 1회의 방향 전환으로 도달할 수 있습니다. 2가지의 경로 중 근수가 존재하지 않는 경로를 택할 수 있기 때문입니다.

B. 문자열 줄이기

`string`

출제진 의도 – **Easy**

- 제출 82번, 정답 17명 (정답률 20.732%)
- 처음 푼 사람: **이재찬**, 9분
- 출제자: 박준영 `sparkjy18`

- 문자열 S를 구성하는 각 문자의 인덱스를 알파벳별로 분류하여 오름차순으로 저장합니다.
- 사전 순으로 앞서는 알파벳부터 순회하며 저장된 인덱스에 위치한 문자를 체크합니다.
  Boolean 자료형의 배열 등을 이용하면 됩니다.
- 체크된 문자를 제외한 나머지 문자를 출력합니다.
- 시간 복잡도는 $O(N)$입니다.

C. 두 스택

greedy, prefix_sum
출제진 의도 – **Easy**

- 제출 90번, 정답 14명 (정답률 15.556%)
- 처음 푼 사람: **김규민**, 18분
- 출제자: 송근수 hulahula3247

- 크게 2 가지 방법으로 풀이할 수 있습니다.
- 우선 누적 합을 이용한 방법입니다.
- 1 번째 배낭에서 i 개의 물건을 제거하기로 마음먹었으면, 2 번째 배낭에서는 $K - i$ 개의 물건을 제거해야 합니다.
- 이는 누적 합으로 $O(1)$ 에 계산할 수 있고 이를 K 번 계산해 보면 되므로, 총 $O(N)$ 의 시간에 문제를 해결할 수 있습니다.

C. 두 스택

- 다음으로는 그리디적인 사고를 이용한 방법입니다.
- 모든 K 번의 시행마다, 무게가 높은 배낭의 물건을 빼면 됩니다.
- 무게가 낮은 배낭의 물건을 제거한 것이 의미가 있으려면, 어차피 다른 배낭의 무게도 그만큼 낮아져야 하고, 그러려면 무게가 높은 배낭의 물건을 언젠가는 빼내야만 합니다. 그러므로 무게가 높은 배낭의 물건부터 차례대로 빼내는 것이 최적입니다.
- 시간 복잡도는 마찬가지로 $O(N)$ 입니다.

D. $A_i \times A_j$

math

출제진 의도 – **Easy**

- 제출 93번, 정답 15명 (정답률 16.129%)
- 처음 푼 사람: **심진희**, 28분
- 출제자: 송근수 hulahula3247

- 우선 연산을 진행할 원소 A_i 하나를 선정해 봅시다.
- $A_i \le 0$ 이라면 A_j 는 가장 작은 값을 뽑는 것이 최적입니다.
- $A_i > 0$ 이라면 A_j 는 가장 큰 값을 뽑는 것이 최적입니다.
- 결론적으로, 우린 연산을 하기로 마음먹었다면 가장 작은 원소 혹은 가장 큰 원소가 반드시 들어감을 알 수 있습니다.

- 가장 작은 원소를 골랐다고 가정하면, 다른 하나의 원소는 남아있는 원소 중 가장 작은 원소를 골라야 최적일 가능성이 있습니다.
- 가장 큰 원소를 골랐다고 가정하면, 다른 하나의 원소는 남아있는 원소 중 가장 큰 원소를 골라야 최적일 가능성이 있습니다.
- 즉 최적해는 가장 작은 원소 2개를 고르거나, 가장 큰 원소 2개를 고르거나, 연산을 진행하지 않는 3가지 경우에서 나옵니다. 각각을 계산해 보고 최댓값을 출력하면 됩니다.

E. gcd 경로

`ad_hoc, number_theory`
출제진 의도 – **Medium**

- 제출 21번, 정답 11명 (정답률 52.381%)
- 처음 푼 사람: **심진희**, 39분
- 출제자: 강효규 `djs100201`

- 단순 bfs 알고리즘으로 정점들 간의 최단 경로를 모두 구하기에는 간선이 너무 많습니다.
- 서로소인 정점들끼리만 간선을 연결하면 $N = 10^6$ 인 경우 $607\,927\,104\,783$ 개의 간선이 그래프에 존재합니다.
- 따라서 다른 방법을 생각해 봅시다.
- 그래프의 특수한 성질을 파악하여 최단 경로가 어떻게 생기는지 파악해 봅시다.

- 만약 $gcd(x, K) = 1$ 이면, $dist(x, K) = 1$ 임을 쉽게 알 수 있습니다.
- 다음으로 1 번 정점과 나머지 모든 정점이 연결되어 있다는 관찰을 할 수 있습니다.
- $x \to 1 \to K$ 와 같이 이동한다면, $dist(x, K) \leq 2$ 입니다.
- 따라서 $gcd(x, K) \leq 2$ 인 x 의 개수를 세주면 됩니다.
- K 가 1 또는 2 일 때를 주의하여 세야 합니다.

F. 서강 피자

greedy, prefix_sum, ad_hoc
출제진 의도 – **Medium**

- 제출 30번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: -, -분
- 출제자: 조원빈 wbcho0504

- 이 문제는 다음과 같이 치환할 수 있습니다.
- M 개의 task가 있습니다.
- i 번째 task는 k_i 개의 subtask를 가지며, 모든 subtask를 t_i 까지 완료해야 합니다. 단, 같은 task에 속하는 subtask들은 동시에 진행할 수 없습니다.
- 이 때 동시에 진행되는 subtask 개수의 최댓값을 최소화해야 합니다.

- i 번째 task에 대해 생각해 봅시다.
- t_i 까지 k_i 개의 subtask를 수행하는 조건은 다음과 같이 치환할 수 있습니다.

  $t_i - k_i + 1$ 까지 적어도 1 개의 subtask를 끝내야 합니다.

  $t_i - k_i + 2$ 까지 적어도 2 개의 subtask를 끝내야 합니다.

  ...

  t_i 까지 적어도 k_i 개의 subtask를 끝내야 합니다.

  단, 같은 task에 속하는 subtask들은 동시에 진행할 수 없습니다.
- 위의 k_i 개의 조건을 만족하도록 subtask를 배치할 수 있다면 task를 완료할 수 있습니다.

- 앞서 설명한 것처럼 각 task의 조건을 나눠줍니다.
- 이제 특정 시점 t에 대해 모든 task의 조건을 만족하는 배치를 찾아야 합니다.
- 시점 t에서 i번 task가 완료해야 하는 subtask의 수를 x_i라고 할 때, 동시에 진행해야 하는 subtask 개수의 최솟값은 $\left\lceil \left(\sum_{i=1}^{M} x_i \right) / t \right\rceil$ 입니다.
- 각 x_i에 대해 $x_i < t$이므로 단순하게 x_1부터 x_M까지의 task를 하나로 이어붙인 후 t마다 나누어 준다면 하나의 task가 동시에 진행되는 경우는 존재하지 않습니다.
- 또한 빈 시간 없이 처음부터 채워 넣었으므로 최적임을 알 수 있습니다.

- 따라서 특정 시점에서 i번 task가 완료해야 하는 subtask의 수인 x_i의 합만 구하면 됩니다.
- task i에 대해 $t_i - k_i + 1$부터 t_i까지, 1부터 k_i까지의 값을 갱신하는 것은 누적 합과 sweeping을 이용해 빠르게 계산할 수 있습니다.
- 최종적으로 시간 복잡도 $O(N)$에 문제를 해결할 수 있습니다.

G. 근수의 미로게임

bfs, game_theory
출제진 의도 – **Hard**

- 제출 4번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: **-**, -분
- 출제자: 송근수 hulahula3247

- 문제를 역순으로 생각하고 bfs를 돌려 봅시다.
- 모든 도착점을 큐에 넣고 bfs를 돌려서 시작점까지 탐색하기 시작합니다.
- 큐에 있는 정점들에 대해서 상하좌우에 붙어있는 정점들을 탐색합니다.
- 이때, 어떠한 정점을 **2 번째**로 방문했을 때 방문 처리를 하고 큐에 넣어 줍니다.
- 왜 1 번째가 아닌 2 번째일까요?

- 어떠한 정점을 1 번째로 방문했을 때는, 실제로는 그 정점으로 가지 못합니다.
- 그 방향은 도착점으로 가는 최단 거리이므로 승형이가 그 방향으로의 이동을 금지할 것이기 때문입니다.
- 그러므로 어떠한 정점을 2 번 방문했을 때 방문 처리를 하는 bfs를 돌려야 합니다.
- bfs를 마쳤을 때 시작점에 저장되어 있는 최단 거리가 정답입니다.

H. 물통

`parametric_search, math`

출제진 의도 – **Hard**

- 제출 0번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: -, -분
- 출제자: 강효규 `djs100201`

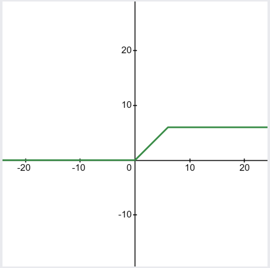
- 크게 2가지 방법으로 풀이할 수 있습니다.
- 1번째는 sparse table 혹은 set과 같은 자료 구조를 사용한 dp 풀이입니다.
- 2번째는 그래프의 개형을 관리하는 구현하기 쉬운 풀이입니다.
- 아이디어 위주로 설명하겠습니다.

- 1번째 풀이의 핵심 아이디어는 각 연산 이후 물의 양에 따라 상태를 3가지로 분할하는 것입니다.

1. 물통에 물이 없는 상태
2. 물통에 물이 꽉 차있는 상태
3. 1번과 2번이 아닌, 물이 적당히 있는 상태

- 1번 상태와 2번 상태를 **종료** 상태라고 합시다.
- 특정한 종료 상태 A 이후 연산을 시행하여 최초로 나오는 종료 상태 B를 생각해 봅시다.
- A와 B 사이의 상태는 누적 합의 합으로 쉽게 표현 가능합니다.
- 그 이후의 최초 종료 상태만이 중요하고, 나머지는 누적 합으로 적절히 확인할 수 있습니다.
- 최초 종료 상태를 찾는 과정에서 sparse table이나 set과 같은 자료 구조를 사용할 수 있습니다.

- 2 번째 풀이는 그래프로 문제를 표현하는 풀이입니다.
- x 축은 시작할 때 물의 양, y 축은 끝날 때 물의 양을 표현하는 그래프를 그려 봅시다.
- 연산을 시작하기 전에는 아래 그림과 같은 그래프로 상태가 표현됩니다.



- 이제 이 상태에서 각 연산에 따라 그래프가 어떻게 변하는지 생각해 봅시다.
- 각 연산의 A_i 는 그래프를 y 축으로 A_i 만큼 평행 이동하는 것과 동치입니다.
- 평행 이동 이후의 그래프는 어떻게 될까요?
- 음수 쪽의 함숫값은 전부 0 이 되고, x 보다 큰 쪽의 함숫값은 전부 x 가 됩니다.

- 따라서 그래프는 다음과 같은 형태를 띄게 됩니다.
- 처음에는 상수 함수의 형태입니다.
- 그 다음에는 기울기가 1 인 직선의 형태입니다.
- 그 후의 특정 지점부터는 다시 상수 함수입니다.
- 결론적으로 기울기의 변화가 생기는 양 끝 점만 알면 $O(1)$ 에 쿼리를 처리할 수 있습니다.
- 이는 parametric search로 쉽게 해결 가능합니다.

Champion

문제		의도한 난이도	출제자
A	SPC에 가는 길	**Easy**	이승형 tmdgud0617
B	직사각형	**Easy**	박수현 shiftpsh
C	두 덱	**Easy**	이승형 tmdgud0617
D	두더지 찾기	**Medium**	송근수 hulahula3247
E	$A_i + A_j$	**Medium**	강효규 djs100201
F	정사각형과 쿼리	**Hard**	조원빈 wbcho0504
G	트리 채우기	**Hard**	조원빈 wbcho0504
H	패널 최적화(Easy)	**Hard**	송근수 hulahula3247
I	패널 최적화(Hard)	**Challenging**	송근수 hulahula3247

A. SPC에 가는 길

ad_hoc
출제진 의도 – **Easy**

- 제출 175번, 정답 42명 (정답률 24.000%)
- 처음 푼 사람: **박성준**, 3분
- 출제자: 이승형 `tmdgud0617`

- 우선 근수가 존재하지 않을 때의 답에 대해 생각해봅시다.
- 승형이의 집과 SPC 대회장의 x 좌표 혹은 y 좌표가 같다면 방향 전환 없이 도달할 수 있습니다.
- 그렇지 않다면, 반드시 1 회의 방향 전환이 필요합니다.
- 이 경우에는 2 회 이상의 방향 전환은 필요하지 않습니다.

- 이제, 근수의 위치에 따른 답의 변화를 살펴봅시다.
- 승형이의 집과 SPC 대회장의 x 좌표 혹은 y 좌표가 같을 때, 근수가 그 사이에 위치하지 않는다면 여전히 방향 전환 없이 도달할 수 있습니다.
- 근수가 승형이의 집과 SPC 대회장 사이에 위치한다면, 2 회의 방향 전환으로 도달할 수 있습니다.
- 기존에 1 회의 방향 전환으로 도달할 수 있었던 SPC 대회장은 근수의 위치와 관계없이 여전히 1 회의 방향 전환으로 도달할 수 있습니다. 2 가지의 경로 중 근수가 존재하지 않는 경로를 택할 수 있기 때문입니다.

B. 직사각형

`math, binary_search`

출제진 의도 – **Easy**

- 제출 76번, 정답 19명 (정답률 25.000%)
- 처음 푼 사람: **하건영**, 11분
- 출제자: 박수현 `shiftpsh`

✓ 다음 조건을 만족하는 크기 $H \times W$ 의 직사각형을 찾아야 합니다.

1. H 와 W 는 정수,
2. 넓이 $HW \geq N$,
3. 둘레 $2(H+W)$ 가 최소.

방법 1 : 미분

- H 가 이미 결정된 값이라고 생각해 봅시다. 이때 $2(H+W)$ 를 최소로 만드려면 W 가 가능한 한 작으면 됩니다.
- 따라서, $HW \geq N$ 이고 W 가 정수임에 따라,

$$W = \left\lceil \frac{N}{H} \right\rceil = \left\lfloor \frac{N+H-1}{H} \right\rfloor$$

로 두면 됩니다.

팁: $\lceil P/Q \rceil = \lfloor (P+Q-1)/Q \rfloor$.

- 이제 H 를 결정할 차례입니다. 다음 식을 최소로 만드는 $[1, N]$ 사이의 값을 찾으면 됩니다:

$$2\left(H + \left\lceil \frac{N+H-1}{H} \right\rceil\right) = 2\left(\left\lceil \frac{H^2+H+N-1}{H} \right\rceil\right).$$

- ceil 함수 안의 값이 최소일 때 전체 식도 최소입니다.

$$\frac{d}{dH}\frac{H^2+H+N-1}{H} = \frac{H^2-N+1}{H^2}$$

이므로, H 가 $\sqrt{N-1}$ 에 가까운 정수일 때 전체 식이 최소입니다.

- 따라서 $H = \mathrm{round}\left(\sqrt{N-1}\right)$ 로 하면 정답입니다.
- $N = 1$ 일 때 $H = 1$ 이어야 함에만 주의하세요!

방법 2 : 기하적인 발상

- 둘레가 $2(H+W)=2X$ 라고 가정합시다. 가로의 길이와 세로의 길이를 더하면 X 입니다.
- X 가 짝수인 직사각형이 가장 넓을 경우는 높이와 너비가 $\dfrac{X}{2}$ 인 경우입니다.
- X 가 홀수인 직사각형이 가장 넓을 경우는 높이가 $\left\lfloor \dfrac{X}{2} \right\rfloor$, 너비가 $X-\left\lfloor \dfrac{X}{2} \right\rfloor$ 인 경우입니다.
- 이 사실을 이용해 최소가 될 수 있는 X 를 이분 탐색으로 구할 수 있습니다.

C. 두덱

`prefix_sum`

출제진 의도 – **Easy**

- 제출 40번, 정답 11명 (정답률 27.500%)
- 처음 푼 사람: **박성준**, 22분
- 출제자: 이승형 `tmdgud0617`

C. 두 덱

- 우선 원빈이의 입장에서는 가방들을 최대한 가볍게 만들어야 하므로, 물건을 K 번 제거해야 합니다.
- 1 번째 배낭에서 i 개의 물건을 제거하기로 마음먹었으면, 2 번째 배낭에서는 $K-i$ 개의 물건을 제거해야 합니다.
- 어떠한 배낭에서 i 개의 물건을 제거하여 $N-i$ 개의 물건을 남기기로 결정한다는 것은, 수열에서 길이가 $N-i$ 인 연속된 구간을 선택하는 것과 같습니다.
- 어떠한 구간에 속한 물건들의 무게의 총합을 **구간의 무게**라고 합시다.
- 1 번째 배낭에서 길이가 $N-i$ 인 연속된 구간을 고르고, 2 번째 배낭에서 길이가 $N-(K-i)$ 인 연속된 구간을 골라 두 구간의 무게 중 최댓값을 최소화하는 것으로 문제를 이해할 수 있습니다.

- 누적 합을 사용하여 각 배낭마다 길이가 i 인 구간의 무게의 최솟값을 구할 수 있습니다.
- 각 구간의 길이 i 마다 시작점 j 를 정하고 누적 합 연산을 통해 해당 구간의 무게를 $O(1)$ 에 구할 수 있습니다.
- 구간의 길이는 최대 N 이고 각 구간마다 가능한 시작점의 개수는 $N - i + 1$ 개이므로, 시간 복잡도는 $O(N^2)$ 입니다.
- 구간의 무게를 전처리한 뒤, 1 번째 배낭에서 길이가 $N - i$ 인 연속된 구간을 고르고, 2 번째 배낭에서 길이가 $N - (K - i)$ 인 연속된 구간을 골라 답을 구할 수 있습니다.
- 총 시간 복잡도는 $O(N^2)$ 입니다.

D. 두더지 찾기

number_theory, ad_hoc
출제진 의도 – **Medium**

- 제출 84번, 정답 10명 (정답률 11.905%)
- 처음 푼 사람: **박성준**, 38분
- 출제자: 송근수 hulahula3247

- 우선 $B_i = 1$ 인 두더지들만 고려해 봅시다.
- 정답 T 는 $B_i = 1$ 인 모든 A_i 들을 약수로 가져야 합니다.
- 이를 만족하는 T 는 $B_i = 1$ 인 모든 A_i 들의 최대공배수(LCM)를 약수로 가져야 합니다.
- LCM 이 L 의 최댓값을 초과하면 정답으로 가능한 시각이 없으니 바로 -1 을 출력해야 합니다.
  연산을 계속 진행하면 오버플로우로 인하여 잘못된 값이 나올 수 있습니다.

D. 두더지 찾기

- 이제 정답은 LCM 혹은 -1 입니다. 왜일까요?
- 어떠한 $B_i = 0$ 인 A_i 에 대하여 LCM 을 A_i 로 나눈 나머지가 0 이면 어떠한 T 도 정답이 될 수 없습니다.
- 가능한 시각 T 는 LCM 의 배수 형태인데, LCM 을 A_i 로 나눈 나머지가 0 이면 LCM 의 배수를 A_i 로 나눈 나머지 역시 0 이기 때문입니다.
- 즉 $B_i = 0$ 인 모든 A_i 에 대하여 LCM 을 A_i 로 나눈 나머지가 0 이 아니면 정답은 LCM, 0 이면 정답은 -1 이 됩니다.

E. $A_i + A_j$

dijkstra, dp

출제진 의도 – **Medium**

- 제출 24번, 정답 5명 (정답률 20.833%)
- 처음 푼 사람: **김동건**, 65분
- 출제자: 강효규 djs100201

- S 와 각 정점의 최단 경로의 길이를 $dist1$ 에 저장합니다.
- T 와 각 정점의 최단 경로의 길이를 $dist2$ 에 저장합니다.
- 정점을 $dist1$ 을 기준으로 정렬해 봅시다.
- 그렇다면 다음과 같은 dp 느낌의 관찰을 할 수 있습니다.

- 새롭게 정렬된 배열을 순회하면서 연결된 정점들의 A의 누적값의 최솟값을 갱신합니다.
- 이 누적된 값과 함께 간선을 확인하면서 최단 경로의 길이를 갱신하면 정답을 구할 수 있습니다.
- $dist1[u] + dist2[v] + c = dist1[T]$를 만족할 때만 최단 경로의 길이를 갱신합니다.
- 다익스트라 알고리즘을 사용하면 정답을 구할 수 있습니다.
- 전체 시간 복잡도 $O(MlogN)$에 문제를 해결할 수 있습니다.

F. 정사각형과 쿼리

`prefix_sum`
출제진 의도 – **Hard**

- 제출 9번, 정답 3명 (정답률 33.333%)
- 처음 푼 사람: **박성준**, 102분
- 출제자: 조원빈 `wbcho0504`

- 매 쿼리마다 크기가 $K \times K$인 정사각형에 포함된 영역의 수를 지웁니다.
- 수를 지우면 당연하게도 서로 다른 수의 개수는 변화하지 않거나 줄어듭니다.
- 그렇다면 서로 다른 수의 개수가 줄어드는 경우에 대해 생각해 봅시다.

- 임의의 수 t가 답에서 제외되기 위해서는 격자에 존재하는 모든 t가 $K \times K$ 크기의 정사각형에 의해 지워져야 합니다.
- 모든 t에 대하여 x, y좌표의 최솟값과 최댓값을 구해 각각 $minX, maxX, minY, maxY$ 라고 합시다.
- $minX$ 와 $maxX$ 를 양 끝 세로, $minY$ 와 $maxY$ 를 양 끝 가로로 하는 직사각형을 그릴 때, $K \times K$ 정사각형이 이 직사각형을 포함해야만 t가 답에서 제외될 수 있습니다.

- 임의의 수 t가 답에서 제외되는 $K \times K$ 정사각형의 왼쪽 위 꼭짓점에 대응되는 칸에 -1을 저장합니다.
- 이제 직사각형 영역의 수를 지우는 연산을 빠르게 하는 방법만 생각하면 됩니다.

F. 정사각형과 쿼리

- 2차원 누적 합을 이용합시다.
- 1부터 $H \times W$까지의 수에 대해 누적 합이 적용되는 직사각형 영역을 구합니다.
- 아래 그림과 같이 색칠된 영역에 누적 합을 적용하는 경우 4개의 칸에 값을 업데이트합니다.

	−1			+1
	+1			−1

F. 정사각형과 쿼리

- 모든 수에 대해 업데이트를 해준 후에 2차원 누적 합을 이용합니다.
- 먼저 모든 행에 대해 누적 합을 적용합니다.

	−1	−1	−1	0
	+1	+1	+1	0

✓ 모든 열에 대해 누적 합을 적용합니다.

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

- 위의 방법과 같이 각 수에 대해 4개의 칸에 업데이트를 해준 후 2차원 누적 합을 이용해 한 번에 연산을 해 줍시다.
- 수는 $H \times W$개이고 2차원 누적 합을 이용하면 $O(H \times W)$에 전처리를 할 수 있습니다.
- 쿼리는 Q개가 주어지고 각 쿼리에 대해서는 $O(1)$에 답을 구할 수 있으므로 시간 복잡도 $O(H \times W + Q)$, 공간 복잡도 $O(H \times W)$에 문제를 해결할 수 있습니다.

G. 트리 채우기

`tree, constructive`

출제진 의도 – **Platinum**

- 제출 16번, 정답 1명 (정답률 6.250%)
- 처음 푼 사람: **박성준**, 118분
- 출제자: 조원빈 `wbcho0504`

- 정점이 N 개인 트리가 주어집니다.
- 각 정점에 1 부터 N 까지의 수를 labeling 해야 합니다.
- 부모 정점의 label이 자식 정점의 label보다 커야 하며, 일부 정점은 이미 labeling이 되어 있습니다.

- 이미 labeling이 되어 있는 정점을 기준으로 segment를 나눠줍니다.
- 그러면 하한값 L과 상한값 R을 갖고 그 사이에 labeling을 해 주어야 하는 정점이 K개인 segment들을 얻을 수 있습니다.
- 즉, 각 segment는 (L, R, K)로 나타낼 수 있습니다.
- 정점에 labeling을 할 때에는 R이 작은 segment를 골라 작은 label을 먼저 labeling을 해주는 것이 이득입니다.

- segment i, j에 대해 $R_i < R_j$ 인 경우를 생각해 봅시다.

1. $L_i < R_i < L_j < R_j$ 인 경우
   - 두 segment에 겹치는 구간이 없으므로 어떻게 배치하든 큰 상관이 없습니다.
2. $L_i < L_j < R_i < R_j$ 인 경우
   - segment i 에서 작은 label부터 채울 때 segment j 에서 사용할 수 있는 label이 가장 많으므로 최적입니다.
3. $L_j < L_i < R_i < R_j$ 인 경우
   - segment j 의 구간이 segment i 의 구간을 포함하므로 segment i 의 labeling을 어떻게 하든 segment j 에서 사용할 수 있는 label의 개수는 같습니다.

- 따라서 segment를 R이 작은 순으로 정렬한 후 각 segment에 대해 L 이상이면서 사용할 수 있는 가장 작은 label을 할당해 줍니다.
- 단, 이 문제는 구간이 아니라 트리를 다루므로 부모 노드의 자식이 여럿일 때 자식을 모두 labeling한 후 부모 노드를 labeling해야 함에 유의합시다.
- 시간 복잡도는 $O(NlogN)$입니다.
- 추가로 루트 노드에서부터 내려오면서 priority queue 등을 이용하는 방식으로도 문제를 해결할 수 있습니다.

H. 패널 최적화 (Easy)

dp, bitmask
출제진 의도 – **Hard**

- 제출 2번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: **-**, -분
- 출제자: 송근수 hulahula3247

H. 패널 최적화(Easy)

- 최대 크기가 10×10인 격자판 위에서의 최적해를 찾으면 됩니다.
- 유명한 백준 문제인 격자판 채우기와 마찬가지로, bitmask를 이용한 dp로 해결할 수 있습니다.
- 최근 N개의 격자판의 상태와 인덱스를 state로 취급해서 관리해 주면 됩니다.
- 다만 가능한 상태가 음수, 0, 양수 총 3가지 이므로, 3 단위로 숫자를 끊어서 관리해 줍니다.

- 격자판들의 상태는 3^N 개 있고 인덱스는 N^2 개 있습니다.
- 하나의 state에서 할 수 있는 행동은, 전압을 음수, 0, 양수로 만들기 이므로 총 3 가지입니다.
- 즉 전체 계산량은 $3^{N+1} \times N^2$ 에 비례합니다.
- 이는 $N = 10$ 일때 약 $1\,700$ 만 정도이므로 충분히 시간 안에 계산할 수 있습니다.
- 별해로 DLAS를 이용한 휴리스틱 기법을 활용해도 통과할 수 있습니다.

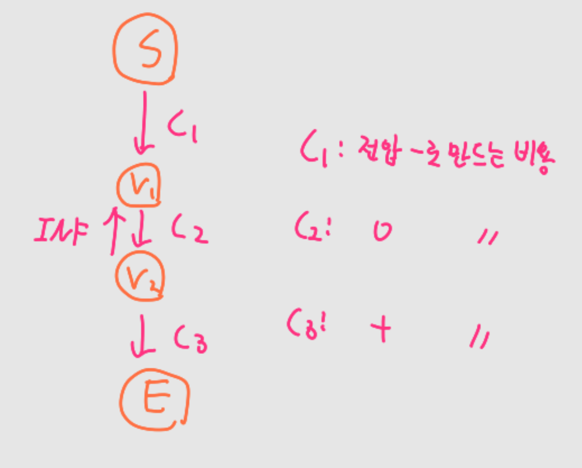
I. 패널 최적화 (Hard)

max_flow_minimum_cut, bipartite_graph
출제진 의도 – **Challenging**

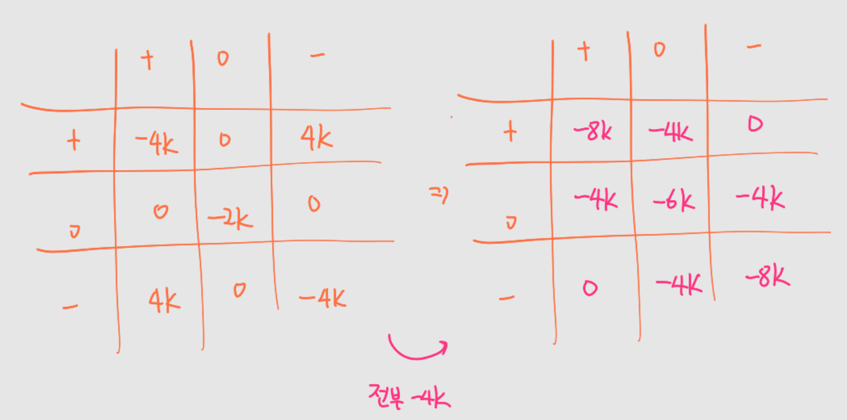
- 제출 3번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: **-**, -분
- 출제자: 송근수 hulahula3247

- N 제한이 커져서 더 이상 dp나 휴리스틱으로는 정답을 구하기 어렵습니다.
- 문제를 flow를 이용한 손해 최소화 모델링으로 설계할 수 있다는 점과 격자판의 상호 작용 상황이 이분 그래프라는 점을 이용해 해결할 수 있습니다.
- 이 문제의 해설은 flow와 max-flow minimum-cut 이론을 안다는 가정하에 작성되었습니다.

- 우선 기본적으로 손해 최소화 모델링을 이용해 봅니다.
- 한 격자에 대해서 3가지 상태가 있을 수 있고, 이를 정점 분할해서 모델링하면 다음과 같습니다.

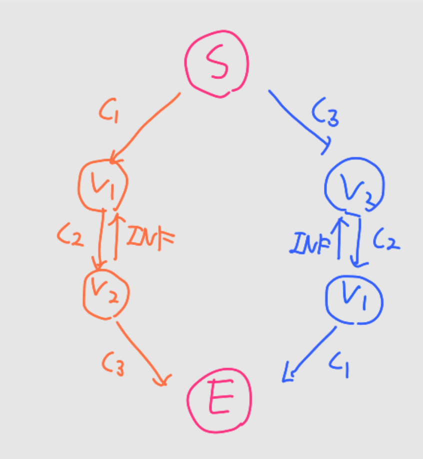


- 격자판의 상호작용 표도 수정이 필요합니다.
- ’손해’ 최소화 모델링이므로 이득이 있으면 flow로 모델링이 안 됩니다. 전부 손해로 바꿔줍니다.

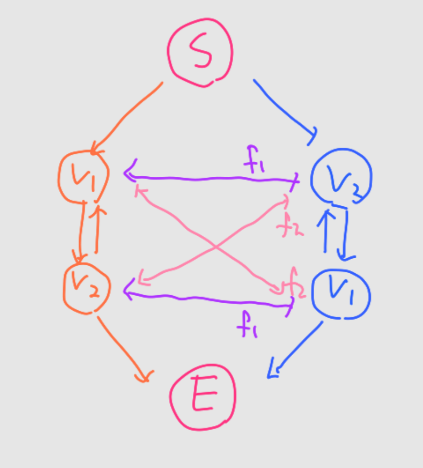


- 기본적으로 max-flow minimum-cut에서는 '서로 다른 속성'을 가지고 있을 때 패널티를 부여할 수 있습니다. 같은 속성일 때는 패널티를 부여하기 힘듭니다.
- 즉 이 문제처럼 같은 전압을 가지고 있을 때 패널티가 더 많이 나오는 상황은 그대로 모델링하면 음수 용량을 가진 모델링이 나옵니다.
- 대신 격자판과 같은 이분 그래프에서는 그래프를 뒤집어서 해결할 수 있습니다.

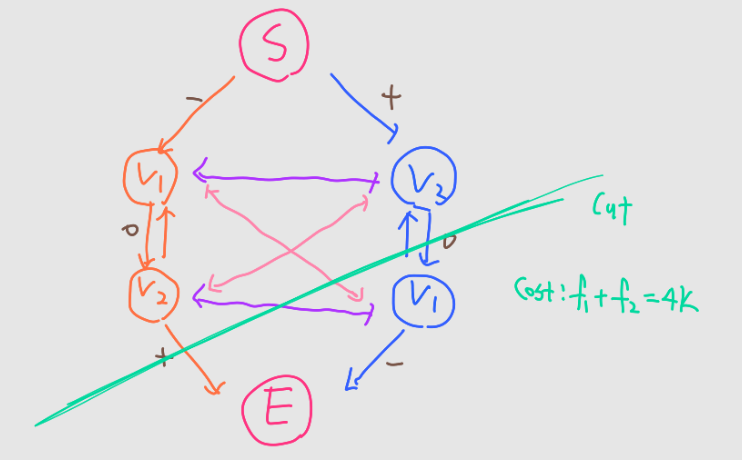
✓ 이분 그래프 한쪽의 정점은 주황색 정점처럼, 나머지 정점은 파란색 정점처럼 모델링하면 여전히 문제를 잘 나타냄을 알 수 있습니다.



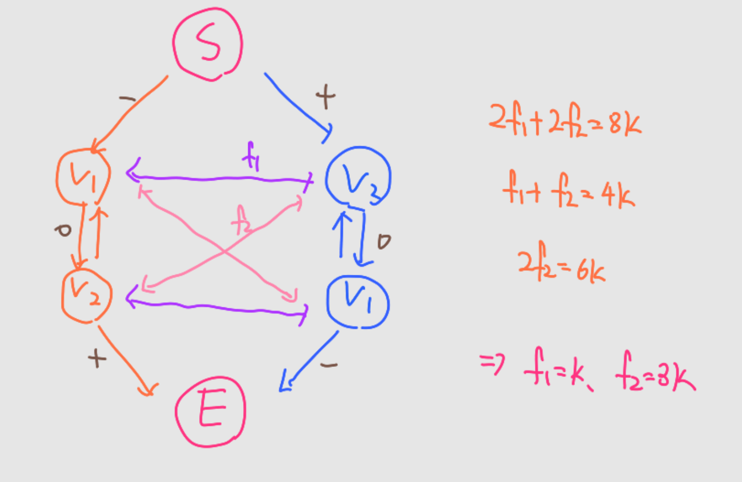
✓ 이제 정점들끼리 간선으로 이어주고, 이 간선들로 표에 있는 손해들을 모델링해야 함을 알 수 있습니다.



- 예를 들어 한 정점에서 양수, 다른 정점에서 0의 전압이 있는 상황을 가정해 봅시다.
- 그래프에서는 다음과 같은 cut이 일어나게 될 것이며 $f_1 + f_2 = 4K$를 만족해야 함을 알 수 있습니다.



- 모든 경우에 대해서 식을 세워 보고 f_1, f_2에 대한 식을 세워 계산해 봅니다.
- $f_1 = K, f_2 = 3K$로 모델링이 잘 됨을 알 수 있습니다.



- 최종적으로 모델링은 다음과 같습니다.
- 문제의 정답은 $4K \times$(격자판 페어 수) $-$ minimum cut이 됩니다.