

UCPC 2022

전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2022

Finals

Official Problemset

본선 문제

2022년 7월 23일, 11:00 → 16:00
스페이스쉐어 삼성COEX센터

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원

SOLVED. AC

ALGO SPOT

MOLOCO



STARTLINK

NAVER D2

한빛미디어
hanbit Media, Inc.

FURIOSA

PLANETARIUM

DEVOCAN

TWIP

programmers

HYUNDAI
AutoEver

NEXON

SCVSOFT

정현환 (LiBe)

문제 목록

문제지에 있는 문제가 총 13문제가 맞는지 확인하시기 바랍니다.

- A 니은숲 예술가
- B NPU 최적화
- C 라즈베리 파이
- D 수열과 쿼리의 부분합의 합
- E 반도체 제작
- F 대충 카드로 몬스터 잡는 게임
- G Traveling Junkman Problem
- H 특별상
- I 사건의 지평선
- J 교집합 만들기
- K 전국 대학생 프로그래밍 대회 동아리 연합 토너먼트
- L 커넥티드 카 실험
- M $\times + \times$

모든 문제의 메모리 제한은 1024 MB로 동일합니다.

문제 A. 니은숲 예술가

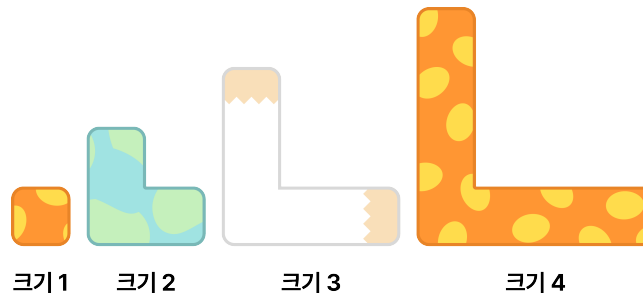
시간 제한 2 초 메모리 제한 1024 MB

루루는 2003년 7월 23일 메이플 월드 빅토리아 아일랜드의 두 마을 “헤네시스”와 “커닝시티” 사이에 있는 “니은숲”이라는 이름의 숲에서 태어났다. 당시 니은숲은 미개척지인데다가 강력한 야생 원숭이 루팡이 서식하고 있어서, 자연경관을 감상하기 위해서 돌아다니는 일부 모험가 외에는 오지 않던 척박한 땅이었다. 그러나 루루의 가족은 니은숲의 낭만적인 경관과 함께 지극정성으로 루루를 키웠다.

어느덧 루루가 자라 초등학교에 입학하게 될 무렵, 니은숲은 빅토리아 아일랜드 재개발 사업으로 인해 흔적도 남기지 않고 폐허가 되었다. 루루의 가족은 적극적으로 저항했지만, 당시 메이플 월드의 대통령 검은 마법사가 군단장을 파견하여 무력으로 밀어붙이자 어쩔 수 없이 커닝시티로 이주하게 되었다.

그때 사건은 루루에게 트라우마로 남게 되었고, 루루는 나중에 무분별한 개발로 메이플 월드가 훼손되더라도 과거 메이플 월드의 아름다움을 전달할 수 있도록 메이플 월드의 다양한 풍경을 가지고 예술 작품을 만들기로 결심하였다. 루루는 오시리아 대륙, 에텔슈타인, 그란디스, 시간의 신전, 아케인 리버 등 다양한 지역을 여행하면서 모험가들에게 각 지역의 풍경을 담은 그림을 그려 주었고, 이윽고 어엿한 스타 예술가가 되었다.

곧 스무 살 생일을 맞는 루루는 생일에 맞춰 특별전을 열기로 했다. 루루가 특별전에서 가장 공을 들여서 만들 조형물은 과거 니은숲의 L 자 지형에서 모티브를 딴 ‘니은 유니버스’이다. 루루는 1부터 N 까지의 서로 다른 크기를 가진 L 자 모양의 조각 N 개를 준비하고, 각 조각에 메이플 월드의 마을 하나를 표현한 조형 예술을 담았다. 크기가 i 인 조각은 크기가 1인 정사각형 $2i - 1$ 개가 가로 i , 세로 i 길이로 붙은 모양이며, C_i 번 마을을 표현하고 있다.



루루는 N 개의 조각을 빈틈없이 붙여서 가로 N , 세로 N 의 정사각형 모양 조형물을 만들려고 한다. 각 조각은 자유롭게 회전할 수 있으며, 같은 마을을 표현한 조각이 붙어 있으면 예술성을 해치기 때문에 이런 조각들은 선분을 공유하면 안 된다.

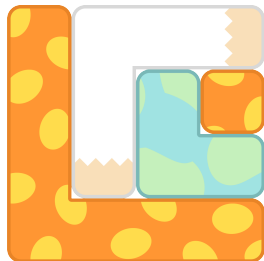


그림 A.1: 4개의 조각으로 조형물을 올바르게 만든 예시



그림 A.2: 같은 마을을 표현하는 조각이 붙어 있어 올바르지 않은 예시

루루는 만들 수 있는 조형물의 후보군을 모두 구한 뒤, 그중에서 가장 조화롭게 보이는 것을 골라 실제 조형물로 제작하려고 한다. 루루가 준비한 조각들의 정보가 주어질 때, 만들 수 있는 서로 다른 조형물의 가짓수를 구하여라. 단, 회전시켜서 같게 만들 수 있는 두 조형물은 같은 조형물이다.

입력

첫 번째 줄에 조각의 수 N 이 주어진다. ($2 \leq N \leq 3000$)

두 번째 줄에 N 개의 정수가 공백으로 구분되어 주어진다. i 번째 수는 크기가 i 인 조각이 표현하는 마을 번호 C_i 이다. ($1 \leq C_i \leq 3000$)

출력

만들 수 있는 조형물의 가짓수를 $998244353 (= 119 \times 2^{23} + 1)$ 으로 나눈 나머지를 출력한다. 998244353 은 소수이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 2 3 1	9

문제 B. NPU 최적화

시간 제한 3 초 메모리 제한 1024 MB

Furiosa AI에서는 인공지능 모델의 학습 및 추론을 기존의 처리 유닛(processing unit)보다 더욱 빠르게 할 수 있는 NPU(Neural Processing Unit)를 만들고 있다. 일반적인 처리 유닛에서 동작하는 프로그램은 호스트(host)에 저장된 데이터를 여러 종류의 연산자를 사용해 처리하여 원하는 값을 계산해 낸다. 이 문제에서는 이 과정을 단순화해, 다음과 같은 환경을 생각한다.

- 호스트에는 1000000개의 데이터를 저장할 수 있는 공간이 있고, 각 공간에는 0번부터 999999번까지의 번호가 붙어 있다.
- 각 연산자는 한 개 이상의 입력 데이터를 받아서 한 개의 출력 데이터를 계산한다. 연산자에도 0 이상 999999 이하의 번호가 붙어 있다.
- 프로그램은 다음과 같은 BNF(Backus-Naur Form)으로 표현된다.

```
<number> ::= 0 | 1 | ... | 999999
<value> ::= <number> | <number>(<list>)
<list> ::= <value> | <list>,<value>
```

- 프로그램은 값 하나를 계산한다. 즉 프로그램은 <value>에 해당하는 문자열이다.
- <value>의 값은 다음과 같이 계산된다.
 - <value>가 <number>로 표현되는 경우, 프로그램의 시작 전 호스트의 <number>번 공간에 저장된 데이터이다.
 - <value>가 <number>(<list>)로 표현되는 경우, <list> 안의 <value>들을 차례대로 입력 데이터로 하여 <number>번 연산자를 사용해 계산한 출력 데이터이다.
- 하나의 프로그램에는 같은 <number>가 여러 번 등장하지 않는다.

Furiosa AI에서 개발하는 NPU는 같은 프로그램을 일반 처리 유닛보다 더 빠르게 실행할 수 있지만, 이를 위해서는 프로그램을 NPU가 사용하는 저수준의 명령어들로 새로 컴파일하는 컴파일러가 필요하다. 같은 프로그램이라도 어떻게 컴파일하느냐에 따라 프로그램의 메모리 사용량이나 실행 시간이 바뀌기 때문에, 효율적으로 NPU를 사용하기 위해서는 최적화된 컴파일러가 필요하다.

NPU에는 M 개의 데이터를 저장할 수 있는 메모리가 있으며, 메모리의 각 공간에는 0번부터 $M-1$ 번까지의 번호가 붙어 있다. NPU는 다음과 같은 세 종류의 명령어를 지원한다.

- $a \gg b$
 - 호스트의 a 번 데이터를 메모리의 b 번 공간에 복사한다. ($0 \leq a < 1000000$; $0 \leq b < M$) 데이터가 이미 있는 경우에는 덮어쓴다.
- $a \ll b$
 - 메모리의 b 번 데이터를 호스트의 a 번 공간에 복사한다. ($0 \leq a < 1000000$; $0 \leq b < M$) 데이터가 이미 있는 경우에는 덮어쓴다.

- $o = w \mid m_1 m_2 \cdots m_l$
 - 메모리의 m_1, m_2, \dots, m_l 주소의 값을 차례대로 입력 데이터로 하는 w 연산자의 출력 데이터를 메모리의 o 공간에 저장한다.
 - 출력 데이터가 저장되는 공간은 입력 데이터가 저장된 공간과 달라야 한다. 즉 $o \neq m_i$ ($1 \leq i \leq l$) 이어야 한다.

NPU 프로그램은 위의 명령어들의 나열으로, 프로그램을 실행하면 프로그램을 이루는 명령어들이 차례대로 실행된다.

프로그램의 효율은 다양한 요인에 의해 결정되지만, 일단 사용하는 명령어 수가 적다면 효율적인 프로그램일 가능성이 높다. <value>에 해당하는 문자열이 주어지면, 명령어를 최대한 적게 사용해 <value>의 값을 계산해서 0 번 메모리에 저장하는 NPU 프로그램으로 변환해 보자.

입력

첫 번째 줄에는 NPU의 메모리의 크기 M 이 주어진다. ($1 \leq M \leq 1000000$)

두 번째 줄에는 계산해야 하는 <value>에 해당하는 문자열이 주어진다. 이 문자열의 길이는 1000000 이하이다.

출력

NPU의 메모리가 부족해서 프로그램을 컴파일할 수 없는 경우에는 -1을 출력한다.

프로그램을 컴파일할 수 있다면 첫 번째 줄에 명령어를 최소한으로 사용해 <value>를 계산하는 프로그램의 명령어 수를 출력한다. 다음 줄부터 프로그램을 구성하는 명령어들을 한 줄에 하나씩 순서대로 출력한다. 한 명령어의 모든 토큰 사이에는 공백이 하나씩 들어간다.

여러 가지가 답으로 가능한 경우 아무거나 하나 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
7 71(72(41,42),73(43,44))	7 41 >> 3 42 >> 4 43 >> 5 44 >> 6 1 = 72 3 4 2 = 73 5 6 0 = 71 1 2
3 71(72(41,42),73(43,44))	9 43 >> 2 44 >> 0 1 = 73 2 0 59 << 1 41 >> 2 42 >> 0 1 = 72 2 0 59 >> 2 0 = 71 1 2
2 71(72(41,42),73(43,44))	-1

문제 C. 라즈베리 파이

시간 제한 2 초 메모리 제한 1024 MB



한별이는 3년 만에 오프라인으로 개최되는 UCPC 본선을 맞아 특별한 이벤트를 계획했다. 바로 참가자들과 라즈베리 파이를 나눠 먹는 것이다! 한별이는 원기둥 모양의 파이를 모든 조각이 밑면이 부채꼴인 기둥 모양이 되도록 M 등분하고, 각 조각 위에 라즈베리를 하나씩 놓았다. 그리고 각 조각에 시계 방향 순서대로 1번에서 M 번까지 번호를 붙였다.

한별이는 대회에 총 $N(N \leq M)$ 명의 참가자가 참가한다는 말을 듣고, 각 참가자에게 몇 번째 조각을 나눠줄지 미리 정해 두었다. 마침내 모든 참가자가 대회장에 도착하고 한별이가 계획대로 파이 조각을 나눠주려는 순간, 한 참가자가 파이 위에 올려진 한 라즈베리를 가리키며 “나에게 저 라즈베리를 주지 않는다면 문제를 10분 만에 다 풀어버리겠다!”라고 선언했다. 그러자 다른 참가자들도 하나둘씩 본인이 원하는 라즈베리를 말하기 시작했고, 결국 모든 참가자가 본인이 먹고 싶은 라즈베리를 하나씩 말하고 돌아갔다.

한별이는 참가자들의 요구를 들어주기 위해 파이에 장식된 라즈베리들의 위치를 조정하려 한다. 그러나 이 라즈베리는 환경 변화에 민감해서 다음과 같은 방법으로 옮기지 않으면 금방 상해 버리고 만다.

- 조각 하나를 선택하여, 그 조각에 있는 모든 라즈베리를 바로 다음 조각으로 옮긴다.

여기서 1번 조각의 바로 다음 조각은 2번 조각, 2번 조각의 바로 다음 조각은 3번 조각, ..., $M-1$ 번 조각의 바로 다음 조각은 M 번 조각, M 번 조각의 바로 다음 조각은 1번 조각이다.

라즈베리는 상하면 다른 라즈베리에도 나쁜 영향을 주므로, 한별이는 어떤 라즈베리도 상하지 않도록 하면서 라즈베리를 최소한으로 옮겨서 모든 참가자의 요구를 들어주려고 한다. 대회가 10분 만에 끝나 버리는 참사를 막기 위해 한별이를 도와주자.

단, 참가자들은 자신이 원하는 라즈베리와 다른 라즈베리를 같이 먹게 되는 것은 신경 쓰지 않으며, 라즈베리를 옮길 때 라즈베리 여러 개가 있는 조각을 선택하더라도 이동 횟수는 한 번으로 친다.

입력

첫 번째 줄에는 파이의 조각 수 M 과 대회 참가자의 수 N 이 공백으로 구분되어 주어진다. ($1 \leq N \leq M \leq 300000$)

두 번째 줄에는 N 개의 정수 a_1, \dots, a_N 이 공백으로 구분되어 주어진다. ($1 \leq a_i \leq M$) a_i 는 i 번째 참가자에게 배정된 파이 조각의 번호를 의미하며, 모든 a_i 는 서로 다르다.

세 번째 줄에는 N 개의 정수 b_1, \dots, b_N 이 공백으로 구분되어 주어진다. ($1 \leq b_i \leq M$) b_i 는 i 번째 참가자가 원하는 라즈베리가 위치한 파이 조각의 번호를 의미한다.

출력

라즈베리를 상하지 않게 하면서 모든 참가자의 요구를 들어줄 수 있으면 라즈베리의 이동 횟수의 최솟값을 출력한다. 그렇지 않은 경우 -1 을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 2 3 5 1 4	3
3 2 3 2 1 2	5
4 3 1 3 4 1 1 3	-1

노트

첫 번째 예제에서, 다음과 같은 방법으로 라즈베리를 총 3번 옮겼을 때 모든 참가자의 요구를 들어줄 수 있다. 이보다 적게 옮기는 방법은 없다.

- i. 1번 조각에 있는 모든 라즈베리를 2번 조각으로 옮긴다.
- ii. 2번 조각에 있는 모든 라즈베리를 3번 조각으로 옮긴다.
- iii. 4번 조각에 있는 모든 라즈베리를 5번 조각으로 옮긴다.

문제 D. 수열과 쿼리의 부분합의 합

시간 제한 4 초 메모리 제한 1024 MB

길이가 N 인 수열 a_1, a_2, \dots, a_N 이 있다. 처음에 모든 a_i 는 0이다. 이때, 다음과 같은 쿼리를 사용해 수열 a 를 변화시킬 수 있다.

$l\ r\ c$: 수열 a 의 l 번째부터 r 번째까지의 값을 c 로 바꾼다.

Q 개의 쿼리가 주어질 때, $f(U, D, L, R)$ 을 ‘ U 번째부터 D 번째까지의 쿼리를 순서대로 사용했을 때 $a_L + a_{L+1} + \dots + a_R$ 의 값’으로 정의한다. f 를 여러 번 시행할 경우, 각 시행은 서로 독립적이라 이전 f 의 시행이 현재 f 의 시행에 영향을 주지 않는다.

$\sum_{U=1}^Q \sum_{D=U}^Q \sum_{L=1}^N \sum_{R=L}^N f(U, D, L, R)$ 을 998244353($= 119 \times 2^{23} + 1$)으로 나눈 나머지를 계산해 보자. 998244353은 소수이다.

입력

첫 번째 줄에 정수 N, Q 가 공백으로 구분되어 주어진다. ($1 \leq N, Q \leq 300000$)

두 번째 줄부터 Q 개의 줄에 걸쳐 Q 개의 쿼리가 한 줄에 하나씩 순서대로 주어진다. i 번째 쿼리로 세 개의 정수 l_i, r_i, c_i 이 공백으로 구분되어 주어진다. ($1 \leq l_i \leq r_i \leq N; 0 \leq c_i < 998244353$)

출력

$\sum_{U=1}^Q \sum_{D=U}^Q \sum_{L=1}^N \sum_{R=L}^N f(U, D, L, R)$ 를 998244353으로 나눈 나머지를 출력하시오.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 2 1 2 1 2 2 2	14
10 10 10 10 593603443 4 9 993565789 3 8 238321270 7 8 424480868 10 10 556869540 8 10 279674600 7 8 575417117 6 8 948583421 6 6 468656456 4 10 865607491	830609277

노트

첫 번째 예제의 경우,

$$f(1, 1, 1, 1) = 1$$

$$f(1, 1, 1, 2) = 1 + 1 = 2$$

$$f(1, 1, 2, 2) = 1$$

$$f(1, 2, 1, 1) = 1$$

$$f(1, 2, 1, 2) = 1 + 2 = 3$$

$$f(1, 2, 2, 2) = 2$$

$$f(2, 2, 1, 1) = 0$$

$$f(2, 2, 1, 2) = 0 + 2 = 2$$

$$f(2, 2, 2, 2) = 2$$

따라서, 답은 $1 + 2 + 1 + 1 + 3 + 2 + 0 + 2 + 2 = 14$ 가 된다.

문제 E. 반도체 제작

시간 제한 4 초 메모리 제한 1024 MB

한별이는 졸업하기 전에 전국 대학생 프로그래밍 대회 동아리 연합에 들어올 후배들을 위해 직접 만든 반도체 몇 개를 기부하려고 한다. 이때 반도체를 최대한 많이 만들기 위해서 반도체 하나를 만드는 데 드는 비용을 최소화하려고 한다.

반도체는 정점이 N 개, 간선이 M 개인 방향 그래프 형태를 하고 있다. 각 정점에는 1번부터 N 번까지의 번호가 붙어 있으며, i 번 정점은 퍼텐셜 에너지 E_i 를 가지고 있다. 퍼텐셜 에너지는 실수 값을 가지며, $E_1 = 1.0$, $E_N = -1.0$ 로 고정되어 있고 나머지 정점의 퍼텐셜 에너지는 한별이가 임의로 정할 수 있다. 덧붙여 1번 정점과 N 번 정점은 특수한 정점이기에 1번 정점으로 들어오는 간선과 N 번 정점에서 나가는 간선은 존재하지 않는다.

반도체를 구성하고 있는 간선 $e = (u, v)$ 은 정점 u 에서 v 로 양의 에너지와 음의 에너지를 각각 전달할 수 있다. 반도체의 각 간선은 에너지 전달 효율이라는 값을 가지고 있다. 만약 한별이가 양의 에너지 전달 효율이 $a_e (\geq 0)$, 음의 에너지 전달 효율이 $b_e (\geq 0)$ 인 간선에 양의 에너지를 양의 실수 $p_e (\geq 0)$, 음의 에너지를 음의 실수 $m_e (\leq 0)$ 만큼 보낸다면, 간선이 전달하는 에너지의 양은 $(a_e p_e + b_e m_e)$ 가 된다. 다만, 간선 $e = (u, v)$ 에 보내는 에너지가 $p_{(u,v)} + m_{(u,v)} \geq E_u - E_v$ 를 만족하지 않으면 과부하로 반도체가 고장날 수 있다.

반도체의 제작 비용은 반도체를 구성하는 각 간선이 전달하는 에너지의 총합과 같다. 좋은 일을 하려는 한별이를 도와, 반도체의 각 정점의 퍼텐셜 에너지와 각 간선에 보내는 에너지의 양을 적절히 조절해 반도체가 고장나지 않으면서 제작 비용이 최소가 되도록 해 보자.

입력

첫 번째 줄에는 정점 개수 N 과 간선 개수 M 이 공백으로 구분되어 주어진다. ($3 \leq N \leq 500$; $1 \leq M \leq N(N-1)$)

두 번째 줄부터 총 M 개의 줄에 걸쳐서 반도체를 구성하고 있는 간선 정보가 공백으로 구분되어 4개의 정수 u, v, a, b 로 주어진다. 이는 u 번째 정점에서 v 번째 정점으로 향하고 양의 에너지 전달 효율이 a , 음의 에너지 전달 효율이 b 인 간선이 있음을 의미한다. 중복 간선이 있는 입력은 주어지지 않는다. ($1 \leq u, v \leq N$; $u \neq v$; $0 \leq a, b \leq 10^9$)

출력

반도체 하나의 제작 비용의 최솟값을 출력하자. 단, 제작 비용이 -3×10^{-9} 보다 작아질 수 있다면, 반도체를 생산할 때마다 돈을 얻을 수 있는 한별이의 기분을 표현하는 단어인 **HAPPY**를 출력하자. 절대/상대 오차는 10^{-9} 까지 허용되며, 답이 -3×10^{-9} 이상 -1×10^{-9} 미만인 입력은 주어지지 않는다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 2 1 2 4 2 2 3 2 1	4.00
3 2 1 2 2 4 2 3 1 2	HAPPY

노트

예제 1은 $p_{1,2} = 0.0$, $m_{1,2} = 0.0$, $p_{2,3} = 2.0$, $m_{2,3} = 0.0$, $E_2 = 1.0$ 로 지정한다면, $p_{1,2} + m_{1,2} = 0.0 \geq E_1 - E_2 = 0.0$, $p_{2,3} + m_{2,3} = 2.0 \geq E_2 - E_3 = 2.0$ 를 만족하게 된다. 비용은 4.0가 되며, 이 이상으로 비용을 줄일 수 없다.

예제 2는 $p_{1,2} = 3.0$, $m_{1,2} = -2.0$, $p_{2,3} = 3.0$, $m_{2,3} = -2.0$, $E_2 = 0.0$ 로 지정한다면, $p_{1,2} + m_{1,2} = 1.0 \geq E_1 - E_2 = 1.0$, $p_{2,3} + m_{2,3} = 1.0 \geq E_2 - E_3 = 1.0$ 를 만족하게 된다. 비용은 -3.0이 되며, 제작 비용이 음수가 될 수 있기에 한별이는 매우 행복해진다.

문제 F. 대충 카드로 몬스터 잡는 게임

시간 제한 2 초 메모리 제한 1024 MB

세계적인 게임 회사 KDH Corp.에서 개발한 턴제 어드벤처 카드 게임인 『대충 카드로 몬스터 잡는 게임』이 드디어 오늘 출시되었다! 다음은 게임의 규칙을 설명한 룰북이다.

- 플레이어는 처음에 1번부터 K 번까지의 카드를 각각 한 장씩 손에 들고 게임을 시작한다.
- 게임은 총 N 개의 턴으로 구성되어 있으며, 각 턴에는 1번부터 K 번까지의 몬스터 중 최대 한 마리가 등장한다.
- 플레이어는 각 턴에 손에 들고 있는 카드 중 최대 2장의 카드를 낼 수 있다. 카드를 한 장도 내지 않고 턴을 넘기는 것도 가능하다.
- 몬스터가 등장한 턴에 플레이어가 등장한 몬스터의 번호와 동일한 번호의 카드를 내면 해당 카드로 몬스터를 처치할 수 있다.
- 만약 어떤 턴에 등장한 몬스터를 그 턴에 처치하지 못하면, 몬스터는 그 턴이 끝난 뒤 도망친다.
- 플레이어가 손에 들고 있는 카드를 모두 소진하고 나면, 턴이 끝난 뒤에 1번부터 K 번까지의 카드를 한 장씩 다시 손으로 들고 온다.
- 더 많은 몬스터를 처치할수록 더 높은 점수를 획득한다.

재야의 게임 고수인 도훈이는 『대충 카드로 몬스터 잡는 게임』이 출시되자마자 1등을 차지했지만, 그의 라이벌 강민이가 그의 1등 자리를 위협하고 있다! 조바심이 난 도훈이는 아예 가능한 최대 점수를 먼저 기록해서 1등을 뺏기는 일을 막고자 한다. 도훈이가 더욱 확실히 1등을 차지할 수 있도록 각 게임에서 처치할 수 있는 최대 몬스터 수를 구해주자.

입력

첫 번째 줄에 게임의 총 턴수 N 과 카드 및 몬스터의 종류 K 가 공백으로 구분되어 주어진다. ($1 \leq N, K \leq 500000$)

두 번째 줄에 각 턴에 등장하는 몬스터의 종류 c_1, c_2, \dots, c_N 이 공백으로 구분되어 주어진다. ($0 \leq c_i \leq K$) $c_i = 0$ 이면 턴 i 에는 몬스터가 등장하지 않았다는 뜻이다.

출력

처치할 수 있는 몬스터 수의 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 4 1 1 2 2 3 3	5
10 5 1 2 2 0 3 3 0 5 4 4	7

노트

첫 번째 입출력 예시에서, 각 턴에 다음과 같은 순서로 카드를 내면 2턴에 등장한 1번 몬스터를 제외한 모든 몬스터를 처치할 수 있다.

- i. 1번 카드와 3번 카드를 내고 1번 몬스터를 처치한다.
- ii. 4번 카드를 낸다. 1번 몬스터는 처치되지 않고 도망간다.
- iii. 2번 카드를 내어 2번 몬스터를 처치한다.
 - 네 장의 카드를 모두 내었으므로 턴 종료 후 카드들을 다시 손으로 들고 온다.
- iv. 1번 카드와 2번 카드를 내고 2번 몬스터를 처치한다.
- v. 3번 카드와 4번 카드를 내고 3번 몬스터를 처치한다.
 - 네 장의 카드를 모두 내었으므로 턴 종료 후 카드들을 다시 손으로 들고 온다.
- vi. 3번 카드를 내어 3번 몬스터를 처치한다.

문제 G. Traveling Junkman Problem

시간 제한 3 초 메모리 제한 1024 MB

고물상이 N 개의 집을 순회하며 물건을 사고 판다. 각 집에는 1번부터 N 번까지 번호가 붙어 있다. 고물상이 취급하는 물건은 총 M 종류가 있으며, 마찬가지로 1번부터 M 번까지 번호가 붙어 있다.

i 번 집은 고물상에게 p_i 가지 서로 다른 종류의 물건을 하나씩 판매하고자 한다. 각 물건의 종류는 $a_{i,1}, a_{i,2}, \dots, a_{i,p_i}$ 번이다. 고물상은 이 중 원하는 물건들만 선택해서 매입할 수 있다.

또한 i 번 집은 q_i 가지 서로 다른 종류의 물건에 관심이 있으며, 각각 $b_{i,1}, b_{i,2}, \dots, b_{i,q_i}$ 번이다. i 번 집은 고물상으로부터 해당하는 종류의 물건들을 몇 개든지 모조리 사들인다. i 번 집이 판매하는 물건들과 i 번 집이 관심을 가지는 물건들의 종류끼리는 서로 겹치지 않는다.

고물상이 j 번 종류의 물건을 매입할 때의 가격은 하나당 s_j , 팔 때의 가격은 하나당 t_j 이다.

고물상은 처음에 아무런 물건도 가지고 있지 않은 상태에서 시작해서, N 개의 집을 원하는 순서로 방문할 수 있다. 단, 각 집은 정확히 한 번씩만 방문해야 한다. 고물상은 순회를 마쳤을 때 수익이 최대가 되는 순서로 집을 방문하려고 한다. 순회를 마치고 남은 물건은 수익에 포함하지 않는다. 얻을 수 있는 최대 수익은 얼마일까?

입력

첫 번째 줄에 N, M 이 공백으로 구분되어 주어진다. ($1 \leq N \leq 18; 1 \leq M \leq 100000$)

두 번째 줄에 고물상이 물건을 매입할 때 드는 비용 s_1, \dots, s_M 이 공백으로 구분되어 주어진다.

세 번째 줄에 고물상이 물건을 판매할 때 버는 수익 t_1, \dots, t_M 이 공백으로 구분되어 주어진다. ($1 \leq s_j < t_j \leq 10^9$)

다음 $2N$ 개 줄에 각 집에 대한 정보가 순서대로 주어진다. i 번 집에 대한 정보는 다음과 같이 두 줄로 이루어진다.

- 첫 번째 줄에 p_i 와 p_i 개의 정수 $a_{i,1}, \dots, a_{i,p_i}$ 가 공백으로 구분되어 주어진다. i 번 집이 판매하는 물건의 종류를 나타낸다.
- 두 번째 줄에 q_i 와 q_i 개의 정수 $b_{i,1}, \dots, b_{i,q_i}$ 가 공백으로 구분되어 주어진다. i 번 집이 관심을 가지는 물건의 종류를 나타낸다.

p_i, q_i 는 0 이상의 정수이며, $0 \leq p_i + q_i \leq M$ 을 만족한다.

각 i 에 대해서 $a_{i,1}, \dots, a_{i,p_i}, b_{i,1}, \dots, b_{i,q_i}$ 는 1 이상 M 이하의 서로 다른 정수이다.

출력

최적의 순서로 N 개의 집을 방문했을 때 얻을 수 있는 최대 수익을 출력한다.

입출력 예시

표준 입력(stdin)

표준 출력(stdout)

```
3 4
2 1 3 4
3 2 5 7
2 2 3
1 4
1 3
2 1 2
2 4 1
0
```

```
5
```

문제 H. 특별상

시간 제한 2 초 메모리 제한 1024 MB

학생 N 명이 미술 대회에 참가하였다. 이 대회에서는 주최자 한 명과 심판 한 명이 수상자를 결정하며, 수상자 결정 방식은 다음과 같다.

1. 주최자와 심판이 각자 모든 학생들의 작품에 점수를 매긴다. 두 사람 모두 점수를 매길 때 서로 다른 두 작품에 같은 점수를 주지 않는다.
2. 주최자가 M 명의 학생을 골라 특별상을 수여한다.
3. 심판은 특별상을 받지 않은 학생들이 그린 작품 중 자신이 매긴 점수가 가장 높은 K 개의 작품을 추리고, 그에 해당하는 K 명의 학생에게 본상을 수여한다.

주최자는 대회에서 종류와 상관 없이 상을 받는 학생들의 작품에 대해 자신이 매긴 점수의 합이 최대가 되도록 하려고 한다. 가능한 합의 최댓값을 구하여라.

입력

첫 번째 줄에 총 학생 수 N , 특별상을 수여할 학생의 수 M , 본상을 수여할 학생의 수 K 가 공백으로 구분되어 주어진다. ($2 \leq N \leq 2 \times 10^5$; $1 \leq M, K \leq N-1$; $M+K \leq N$)

두 번째 줄부터 N 개의 줄에 걸쳐 각 작품에 대해 주최자가 매긴 점수 a_i 와 심판이 매긴 점수 b_i 가 공백으로 구분되어 주어진다. ($0 \leq a_i, b_i \leq 10^9$) 점수는 모두 정수이며, $i \neq j$ 에 대해 $a_i \neq a_j$, $b_i \neq b_j$ 를 만족한다.

출력

상을 받는 $M+K$ 명의 학생이 그린 작품에 대해 주최자가 매긴 점수의 합의 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
7 2 3	33
4 7	
7 8	
2 1	
9 3	
6 0	
10 4-	
3 6	

노트

주최자가 첫 번째와 네 번째 학생을 골라서 특별상을 줄 경우 심판은 자신이 매긴 점수에 따라 두 번째, 여섯 번째, 일곱 번째 학생에게 상을 주게 된다. 이때 상을 받은 5명의 작품에 대해 주최자가 매긴 점수의 합은 33이 되고, 이것이 가능한 최댓값임을 증명할 수 있다.

문제 I. 사건의 지평선

시간 제한 3 초 메모리 제한 1024 MB

전국 대학생 프로그래밍 동아리 연합이 드디어 우주 탐사선 UCPC 1호를 발사했다! 이 탐사선은 알고리즘 문제 풀이의 상징인 수열과 쿼리를 전광판에 표시한 채 우주를 여행하고 있다.

전광판은 N 개의 칸으로 나누어져 있고, 각 칸에는 하나의 정수가 표시되어 있다. 이 수열은 매일 자정에 다른 수열로 바뀌는데, i 번째 칸에는 전날 $l_i, l_i + 1, \dots, r_i$ 번째 칸에 표시되어 있던 수 중 가장 큰 값이 표시된다.

아쉽게도, UCPC 1호는 탐사를 성공적으로 마치지 못하고 블랙홀의 사건의 지평선을 지나 특이점으로 빨려 들어가고 있다. 특이점에 도달하는 순간, 전광판의 각 칸에는 무한한 시간이 흐를 때 그 칸에 무한 번 표시되는 수들 중 가장 큰 수가 표시된다.

특이점에 도달한 UCPC 1호의 전광판의 모습을 구해보자.

입력

첫 번째 줄에는 칸의 개수 N 이 주어진다. ($1 \leq N \leq 300000$)

두 번째 줄에는 각 칸에 최초로 표시된 수 a_1, \dots, a_N 이 공백으로 구분되어 차례로 주어진다. ($1 \leq a_i \leq N$; 모든 a_i 는 정수)

세 번째 줄부터 N 개의 줄에 걸쳐 l_i, r_i 가 공백으로 구분되어 차례로 주어진다. ($1 \leq l_i \leq r_i \leq N$)

출력

UCPC 1호가 특이점에 도달했을 때, 전광판의 각 칸에 표시되는 수를 차례로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4	4 3 3 4
1 2 3 4	
3 4	
3 3	
2 3	
1 2	

노트

표시되는 수열은 차례로 $[1, 2, 3, 4]$, $[4, 3, 3, 2]$, $[3, 3, 3, 4]$, $[4, 3, 3, 3]$, $[3, 3, 3, 4]$, $[4, 3, 3, 3]$, ... 이다.

문제 J. 교집합 만들기

시간 제한 2 초 메모리 제한 1024 MB

구간 $[l, r]$ 이란, l 이상 r 이하의 모든 실수로 이루어진 집합을 의미한다.

구간 N 개가 주어진다. 이때, 다음과 같은 질의 Q 개를 해결하는 프로그램을 작성하시오.

- 주어진 l 과 r 에 대해, 구간을 1개 이상 선택하여 그 교집합이 정확히 $[l, r]$ 이 되도록 할 수 있는가? 만약 가능하다면, 최소 몇 개의 구간을 선택해야 하는가?

입력

첫 번째 줄에는 구간의 개수 N 이 주어진다. ($1 \leq N \leq 300000$)

그 다음 줄부터 N 개의 줄에 걸쳐, 한 줄에 하나씩 구간 $[l_i, r_i]$ 를 나타내는 정수 l_i 와 r_i 가 공백으로 구분되어 주어진다. ($0 \leq l_i < r_i \leq 10^6$)

그 다음 줄에는 질의의 개수 Q 가 주어진다. ($1 \leq Q \leq 300000$)

그 다음 줄부터 Q 개의 줄에 걸쳐, 한 줄에 하나씩 질의에서 주어지는 두 정수 l 과 r 이 공백으로 구분되어 주어진다. ($0 \leq l < r \leq 10^6$)

출력

각 질의마다 한 줄에 하나씩, 구간의 교집합이 정확히 $[l, r]$ 이 되도록 할 수 없으면 -1 을 출력하고, 할 수 있으면 선택해야 하는 구간의 최소 개수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3	2
0 10	-1
2 6	-1
4 8	
3	
4 6	
2 8	
1 9	

문제 K. 전국 대학생 프로그래밍 대회 동아리 연합 토너먼트

시간 제한 2 초 메모리 제한 1024 MB

전대프연은 3년 전에 UCPC를 토너먼트 방식으로 바꾸겠다고 선언한 적이 있지만, 너무 큰 변화를 감행하느라 많은 시행착오를 겪었다. 올해 전대프연의 회장을 맡은 헤아는 내부 문서들을 읽다가 이 대회의 준비 기록을 발견하고, 토너먼트 방식의 대회를 다시 열어 보기로 결심했다.

운이 좋게도 이번 UCPC에는 정확히 2^N 명의 참가자가 참가했기 때문에, 헤아는 싱글 엘리미네이션 방식으로 아주 깔끔하게 대회를 진행할 수 있게 되었다. 이는 일반적으로 ‘토너먼트’라고 할 때 가장 먼저 떠오르는 대회 방식으로, 과정을 자세히 설명하면 다음과 같다.

1. 참가자들에게 1번부터 2^N 번까지의 서로 다른 시드 번호를 부여하고, 임의의 순서로 한 줄로 나열한다.
2. 줄에 속해 있는 참가자들을 앞에서부터 두 명씩 짝을 짓는다. 서로 짝지어진 두 참가자는 일대일 경기를 진행하고, 여기서 패배한 참가자는 줄에서 떠난다. 이렇게 모든 짝이 경기를 진행하고 나면 줄에 남는 참가자 수는 이전의 절반이 됨을 알 수 있다.
3. 따라서 2번 과정을 N 번 반복하면 단 한 명의 참가자만이 남게 되며, 이 참가자가 대회의 우승자가 된다.

싱글 엘리미네이션 토너먼트는 흔히 아래와 같은 이진 트리 모양의 대진표로 나타내어진다.

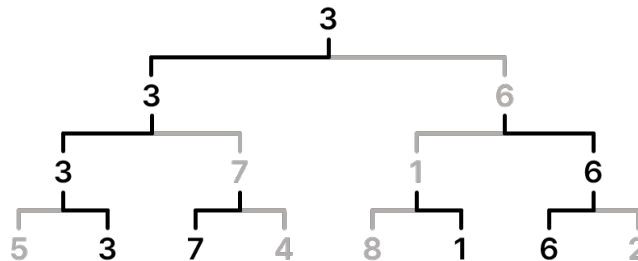


그림 K.1: $2^3 = 8$ 명의 참가자가 참가한 대회의 대진표의 예시

헤아를 포함한 대회 운영진은 대회에서 치러진 모든 경기를 감독하고, 경기 결과를 공유 문서에 기록했다. 그러나 여럿이서 경기 결과를 마구 적어 넣는 바람에 이 기록으로는 대회의 흐름을 전혀 알아볼 수 없게 되었다. 게다가 대회가 끝나고 경기 수를 세어 본 운영진은 경기 하나가 누락되었다는 절망적인 사실을 발견했다.

대회를 진행하느라 너무 지쳐서 이 사태를 수습할 여력이 없는 운영진을 대신해, 기록에서 누락된 경기 하나를 찾아 주자.

입력

첫 번째 줄에 정수 N 이 주어진다. ($2 \leq N \leq 20$)

다음 $2^N - 2$ 개의 줄에 걸쳐 각 줄에 각 경기의 결과를 나타내는 두 개의 정수 a_i 와 b_i 가 공백으로 구분되어 주어진다. 이는 경기에서 a_i 번 참가자와 b_i 번 참가자가 맞붙어 a_i 번 참가자가 이겼음을 의미한다.

주어지는 입력은 올바른 전체 경기 기록에서 경기 하나가 누락된 기록임이 보장된다. 즉, 누락된 경기의 결과를 어떻게 추측하더라도 올바른 기록이 만들어지지 않는 입력은 주어지지 않는다.

출력

누락된 경기의 결과를 입력과 같은 형식으로 두 개의 정수로 출력한다. 가능한 경기 결과가 여러 가지일 경우는 모든 가능한 답을 한 줄에 하나씩 출력한다. 이때 승자의 번호가 작은 것부터, 그런 것이 여러 가지라면 그중 패자의 번호가 작은 것부터 출력하도록 한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3	6 2
3 6	
1 8	
3 7	
3 5	
6 1	
7 4	
2	1 3
3 4	3 1
1 2	

노트

첫 번째 예제의 대진표는 지문에서 주어진 그림과 같다.

두 번째 예제는 $2^2 = 4$ 명이 참가한 대회에서 결승전이 누락된 기록으로, 결승전의 승자가 누구인지 알 수 없으므로 두 가지 가능성을 모두 출력한다.

문제 L. 커넥티드 카 실험

시간 제한 2 초 메모리 제한 1024 MB

정보통신기술(ICT)의 발달에 힘입어, 미래형 자동차로 여겨졌던 인터넷 연결을 통해 운전자에게 다양한 서비스를 제공하는 커넥티드 카(connected car)가 현실로 다가왔다. 현대오토에버는 이에 발맞춰 클라우드와 사물 인터넷을 비롯한 최신 ICT를 적용한 차세대 커넥티드 카 서비스 플랫폼을 구축하고, 최고의 커넥티드 카를 완성하기 위한 핵심 소프트웨어 기술을 축적하고 있다.

현대오토에버의 엔지니어 현오는 새로운 서비스를 생각하던 중, 커넥티드 카의 핵심 기술인 사물 인터넷과 위치 기반 기술을 접목한 실험을 해 보기로 했다. 현오가 개발한 실험용 프로그램은 다음과 같은 기능을 가지고 있다.

- 현오는 사물 인터넷에 연결된 커넥티드 카를 원격 조종할 수 있다.
- 사물 인터넷에 연결된 커넥티드 카가 그렇지 않은 커넥티드 카와 같은 위치에 있게 되면, 그 커넥티드 카를 사물 인터넷에 연결시킬 수 있다. 이후 두 커넥티드 카가 다시 서로 멀어지더라도 연결은 그대로 유지된다.

현오는 실험을 위해 1부터 N 까지 번호가 매겨진 N 대의 커넥티드 카를 일렬로 배치했다. i 번 커넥티드 카의 초기 위치는 x_i 이고, 연료량은 h_i 이다. 모든 커넥티드 카는 1만큼의 연료를 소비해서 1의 거리만큼 이동할 수 있으며, 연료를 모두 소비하면 더 이상 움직일 수 없다.

처음에 커넥티드 카들은 모두 사물 인터넷에 연결되지 않은 상태이다. 현오는 먼저 S 번 커넥티드 카를 사물 인터넷에 연결시키고, 프로그램의 기능을 적절히 사용해 다른 커넥티드 카들에 사물 인터넷을 전파하려고 한다.

현오가 커넥티드 카들을 어떻게 다루느냐에 따라 실험에서 사물 인터넷에 연결되는 커넥티드 카들의 조합은 달라질 수 있다. 현오가 다양한 방법으로 여러 번 실험을 진행할 때, 사물 인터넷에 연결될 가능성이 있는 커넥티드 카를 전부 구해 보자.

입력

첫 번째 줄에 N 과 S 가 주어진다. ($1 \leq N \leq 1000000$; $1 \leq S \leq N$)

두 번째 줄에 각 커넥티드 카의 초기 위치 x_1, x_2, \dots, x_N 이 공백으로 구분되어 차례로 주어진다. ($0 \leq x_i \leq 10^9$; $x_i \leq x_{i+1}$)

세 번째 줄에 각 커넥티드 카의 연료량 h_1, h_2, \dots, h_N 이 공백으로 구분되어 차례로 주어진다. ($1 \leq h_i \leq 10^9$)

출력

첫 번째 줄에 사물 인터넷에 연결될 가능성이 있는 모든 커넥티드 카의 번호를 오름차순으로 정렬하여 출력한다.

입출력 예시

표준 입력(stdin)

표준 출력(stdout)

```
5 3
1 2 4 5 8
2 1 2 2 3
```

```
1 2 3 4
```

노트

예제에서 실험 결과로 나올 수 있는 사물 인터넷에 연결된 커넥티드 카들의 조합은 {1,2,3}, {2,3}, {3}, {3,4}가 있다.

문제 M. $\times + \times \times$

시간 제한 4 초 메모리 제한 1024 MB

칠판에 N 개의 정수가 적혀 있다. A_k 와 B_k 를 다음과 같이 정의하자.

- A_k : 칠판에 임의의 두 수를 골라 지우고, 두 수의 곱을 칠판에 쓰는 작업을 k 번 한다. 이때, 칠판에 적힌 수를 모두 **합한** 값의 기댓값.
- B_k : 칠판에 임의의 두 수를 골라 지우고, 두 수의 합을 칠판에 쓰는 작업을 k 번 한다. 이때, 칠판에 적힌 수를 모두 **곱한** 값의 기댓값.

임의로 두 수를 고를 때 모든 쌍이 선택될 확률은 같으며, 모든 시행은 독립이다.

A_0, \dots, A_{N-1} 과 B_0, \dots, B_{N-1} 을 $998244353 (= 119 \times 2^{23} + 1)$ 으로 나눈 나머지를 구하여라. 998244353은 소수이다.

입력

첫 번째 줄에 N 이 주어진다. ($1 \leq N \leq 200000$)

두 번째 줄에 칠판에 적힌 N 개의 정수가 공백으로 구분되어 주어진다. 각 수는 0 이상 998244353 미만이다.

출력

첫 번째 줄에 A_0, \dots, A_{N-1} 을 998244353으로 나눈 나머지를 공백으로 구분하여 출력한다.

두 번째 줄에 B_0, \dots, B_{N-1} 을 998244353으로 나눈 나머지를 공백으로 구분하여 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3	18 39 162
3 6 9	162 66 18

노트

유리수를 기약분수로 나타냈을 때 $\frac{a}{b}$ 인 경우 이 수를 소수인 p 로 나눈 나머지는 $a \equiv c \cdot b \pmod{p}$ 를 만족하는 0 이상 p 미만의 정수 c 이며, b 가 p 의 배수가 아니라면 이 값은 유일하다.

이 문제에서는 가능한 모든 입력에 대해 A_0, \dots, A_{N-1} 과 B_0, \dots, B_{N-1} 이 유리수이고 각 수를 기약분수로 나타냈을 때 분모가 998244353의 배수가 아니라는 것을 증명할 수 있다.