|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | 문제: **(중상) [연습P-0022] LIS 2** | |

|  |  |
| --- | --- |
| 시간 제한 | **10**개의 Test Input 입력시 C/C++ **1.5**초 / Java **2.5**초 |
| 메모리 제한 | Stack : **1** Mbytes  /  Total : **256** Mbytes |
| 제출횟수 제한 | **제한 없음** |
| 채점 | 답안을 제출하면 Test Input에 대한 결과를 판정해서 실시간으로 알려주며  그 의미는 다음과 같다.  **Pass : Test Input에 대하여 모두 정답 처리된 경우**  **Fail : Test Input에 대하여 일부 혹은 모두 오답 처리된 경우**  ※ Test Input : 검정 시스템에서 제출된 코드 실행 時 자동으로 입력되는 Input Data |
| 평가기준 | 정답 처리된 Test Input의 개수 |

**[문제]**  
부분 수열(Subsequence)은 어떤 수열에서 순서를 유지하되, 그 중 일부 항만을 선택하여 만들 수 있는 수열이다. 예를 들어, (1, 3, 2, 4)로 이루어진 수열이 있다면, (1, 3, 4), (1, 2, 4)등은 부분 수열이 될 수 있지만, (1, 2, 3)은 부분수열이 될 수 없다.  
최장 증가 부분 수열(Longest Increasing Subsequence)은 어떤 수열의 부분 수열 중 각 항이 이전 항에 비해 증가하는 부분 수열을 의미한다. 예를 들어, 수열 (1, 8, 4, 12, 2, 14, 6)의 최장 증가 부분 수열은 (1, 8, 12, 14), (1, 4, 12, 14)가 있다.  
수열이 주어졌을 때, 해당 수열의 최장 증가 부분 수열의 길이를 알아내자.

**[입력]**  
첫 번째 줄에 테스트 케이스의 개수 T가 주어진다. 이어서 T개 테스트 케이스의 입력값이 주어진다. 각 테스트 케이스는 두 줄로 구성이 되며 첫 번째 줄에 수열의 길이 N이 주어진다. (1 ≤ N ≤ 300,000)  
두 번째 줄에 수열의 각 항의 값이 순서대로 주어진다. 주어지는 숫자는 32비트 정수형 이내의 숫자이다.

**[출력]**  
각각의 테스트 케이스에 대하여 #x(x는 케이스 번호를 의미)를 출력하고 공백을 하나 둔 다음 주어진 수열의 최장 증가 부분 수열의 길이를 출력한다.

**[입출력 예]**  
(입력)  
1  
10  
1 1 2 2 3 3 2 2 5 5

(출력)  
#1 4

(sample\_input.txt의 출력)  
#1 4  
#2 100  
#3 105

|  |  |
| --- | --- |
| 문제: **(중상) [기출P-0007] 팀 구성** |  |

|  |  |
| --- | --- |
| 시간 제한 | **40**개의 Test Input 입력시 C/C++ **1**초 / Java **2**초 |
| 메모리 제한 | Stack : **1** Mbytes  /  Total : **256** Mbytes |
| 제출횟수제한 | **제약없음** |
| 채점 | 답안을 제출하면 Test Input에 대한 결과를 판정해서 실시간으로 알려주며  그 의미는 다음과 같다.  **Pass : Test Input에 대하여 모두 정답 처리된 경우**  **Fail : Test Input에 대하여 일부 혹은 모두 오답 처리된 경우**  ※ Test Input : 검정 시스템에서 제출된 코드 실행 時 자동으로 입력되는 Input Data |
| 평가기준 | 정답 처리된 Test Input의 개수 |

올림픽에 출전한 N명의 육상 선수가 좌우 방향으로 늘어서 있다. 각 선수의 실력은 정수로 표시된다. 왼쪽부터 순서대로 선수들의 실력을 표시한 배열을 A[1..N]이라고 하자. 올림픽 주최측이 갑자기 원래 개인 종목이던 육상을 팀 종목으로 바꾸었다. 각 선수는 현재 늘어서 있는 위치에서 자기자신을 포함하는 연속된 선수들을 모아서 팀을 만들 수 있다고 한다. 단, 연속된 선수들 중 자기 자신이 아닌 한 명을 제외할 수 있다. 각 선수는 자신을 포함하는 팀의 실력의 합이 가장 큰 것을 원한다. 즉, i번째 선수가 원하는 최선의 팀은 1 ≤ s ≤ i ≤ t ≤ N을 만족하는 s와 t로 A[s]+A[s+1]+…+A[t] 의 값 혹은 A[s]+A[s+1]+…+A[t]에서 A[i]가 아닌 것 중 하나를 제외한 값이 가장 큰 경우이다. 어떤 선수에 대해서는 가능한 가장 큰 값이 음수가 될 수도 있음에 주의하라.

N 명의 실력을 입력으로 받아 각 선수에 대해서 최선의 팀을 구성하는 방법을 계산하는 프로그램을 작성하라. 각 선수에 대해 독립적으로 계산하는 것임에 주의하라. 즉, 어느 한 선수가 구성한 팀이 다른 선수가 구성하려는 팀에 영향을 주지 않는다.

**[입력]**첫째 줄에 테스트 케이스의 개수 T가 주어지고, 이후 차례로 T개 테스트 케이스가 주어진다. 각 케이스의 첫째 줄에는 선수의 수 N이 주어진다. (1 ≤ N ≤ 200,000) 다음 줄에는 각 선수의 실력을 나타내는 N개의 정수가 선수들이 좌우로 늘어선 순서대로 주어진다. 선수들의 실력을 나타내는 값은 -10,000 이상 10,000 이하의 정수이다.

**[출력]**각 테스트 케이스의 답을 순서대로 표준출력으로 출력하며, 각 케이스마다 줄의 시작에 “#x”를 출력하여야 한다. 이때 x는 케이스의 번호이다. 같은 줄에, 테스트 케이스에 대해서 각 선수들에 대해 실력의 최대 합을 계산하여 그 절대값을 우선 취한 후, 모두 더한 값을 1,000,000,007로 나눈 나머지를 출력한다. 어떤 선수에 대한 **최대 합이 음수인 경우 그 절대값을 취해야 함**을 다시 강조한다.

**[입출력 예]**(입력)  
3  
5  
-1 2 -3 2 -1  
1  
0  
4  
1 1 1 1

(출력)  
#1 15  
#2 0  
#3 16

(sample\_input.txt에 대한 출력)

#1 66809074  
#2 49389523  
#3 67001805  
#4 60149837  
#5 68842812  
#6 155404851  
#7 998738202  
#8 449952665  
#9 755501369  
#10 304425242

**[힌트]**

#1 테스트 케이스에 대한 설명

1번 선수 : {-1, 2, -3(제외), 2} -> 합 3  
2번 선수 : {2, -3(제외), 2} -> 합 4  
3번 선수 : {2, -3, 2} -> 합 1  
4번 선수 : {2, -3(제외), 2} -> 합 4  
5번 선수 : {2, -3(제외), 2, -1} -> 합 3

각 선수가 위와 같이 팀을 구성하는 경우, 각 선수가 구성할 수 있는 팀의 최대합의 절대값의 합은  
( |3| + |4| + |1| + |4| + |3| ) % 1000000007 = 15

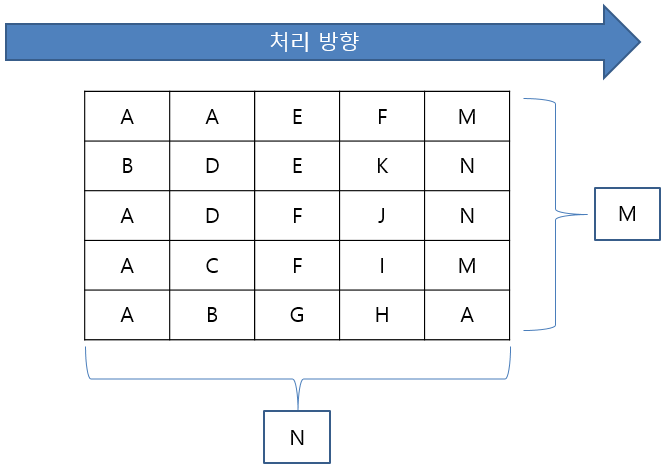
문제: **(중) [사전A-0033] [2019년 12월 PRO 검정 사전 테스트] 스마트 팩토리**

**※ 본 테스트는 Professional 사전테스트로   
   문제풀이 과정에서 본인의 힘으로 문제를 풀이하지 않고 타인의 조언을 구하거나             
   본인이 직접 작성하지 않은 자료를 그대로, 혹은 위조, 변조, 참조하여 제출하는 경우 부정행위로   
   판단될 수 있으며, 이러한 경우 6개월간 모든 SW검정 응시가 제한됨을 알려드립니다.   
   (이미 Professional 이상을 취득한 자격자 또한 본 규정은 동일하게 적용됩니다.)   
  
※ 본 문제는 2019년 11월 14일(목) 13시까지(서버시간 기준)만 풀이가능하며,   
   위 기한 내에 문제의 정답을 제출해야 2019년 12월 Professional SW검정을 신청할 수 있습니다.   
  
※ 마지막 제출 답안 결과가 최종결과로 반영되니, 답안제출에 유의하여 주시기 바랍니다.   
  
※ 11월 14일(목) 13시 이후에도 문제는 계속 오픈되며, 일반 연습문제와 같이 계속 풀이할 수 있습니다.     
   11월 14일(목) 13시 이전에는 정답자이더라도 타 정답자 코드 조회가 불가능하며   
   이후에는 정답자에 한해 타 정답자 코드 조회가 가능합니다.**

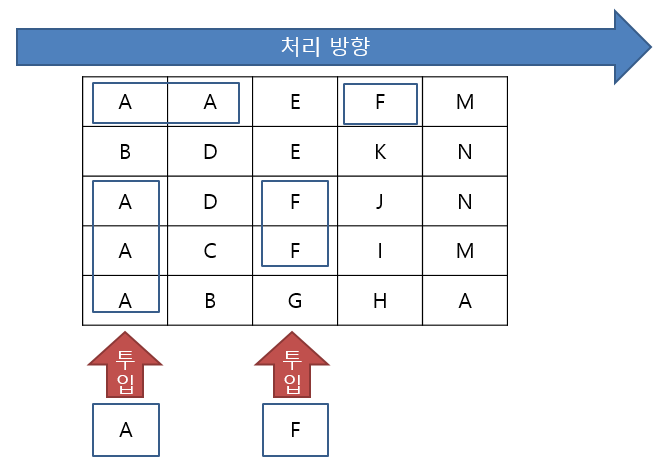
|  |  |
| --- | --- |
| 시간 제한 | **40** 개의 Test Input 입력시 C/C++ **1** 초, Java **1** 초 |
| 메모리 제한 | Stack : **1** Mbytes  /  Total : **256** Mbytes |
| 코드길이 제한 | **128** Kbytes |
| 채점 | 답안을 제출하면 Test Input에 대한 결과를 판정해서 실시간으로 알려준다.  모든 케이스에 대해 정답일 경우 **Pass**, 그 외의 경우 **Fail** 이다. |

A Company는 전자부품을 납품하는 회사이며 중국에 공장을 가지고 있다.   
이 공장에서는 26 가지의 부품을 생산할 수 있으며 각 부품은 기호 A ~ Z 로 나타낸다.   
이 공장의 생산라인은 다음과 같이 구성되어 있다.   
각 라인에서는 하루에 N 개의 부품을 순차적으로 처리 가능하며, 공장 전체에 M 개의 라인이 있고 모든 라인은 동시에 작동을 시작하고, 동시에 작동을 마감한다.   
각 라인의 처리속도는 같으며 모든 부품을 만드는 시간은 동일하다.   
  
그러나 이 공장에는 치명적인 결함이 있어서 엔지니어의 개입이 없을 경우 만들어진 모든 부품이 불량품이 되고 만다.   
이러한 상황을 임시적으로 조치하기 위해서 A Company에서는 K 명의 엔지니어를 파견했다.   
각 엔지니어들은 한 종류의 부품만을 책임지고 담당하며, 투입된 시점부터 퇴근 전까지 모든 라인에서 생산되는 담당 부품의 불량률을 0%로 만들 수 있다.   
엔지니어들은 **한 번 투입되면 공장이 끝날 때까지 일할 수 있지만, 한 번 일한 엔지니어는 그날 다시 일하지 않는다.**   
또한 **두 명 이상의 엔지니어가 동시에 일할 수는 없다.** 즉 다른 엔지니어를 투입하고 싶으면 이전에 있던 엔지니어를 퇴근시키고 교체해야 하며 퇴근한 엔지니어는 다음날 출근한다.

이 공장의 일일 생산계획 및 근무 가능한 엔지니어 리스트를 받았을 때 **엔지니어를 효과적으로 투입하여 생산할 수 있는 정상부품의 최대개수**를 구하라.

  
[그림 1]

[그림 1] 과 같은 일일 생산계획이 주어지고 근무 가능한 엔지니어가 3 명이고 각각 담당하는 부품이 F, G, A인 경우를 생각해보자.

  
[그림 2]

A를 담당하는 엔지니어와 F를 담당하는 두 명의 엔지니어를 [그림 2]와 같이 투입하여 작업을 진행한다면 총 8개의 정상부품을 생산할 수 있으며, 이 경우 생산할 수 있는 정상부품의 수가 최대 값이 된다.

**[제한사항]**

1. N 은 1 이상 5000 이하의 자연수, M 은 1 이상 100 이하의 자연수이다.   
2. K 는 1 이상 5 이하의 자연수이다.   
3. K 명의 엔지니어를 모두 투입 시킬 필요는 없다.   
4. K 명의 엔지니어 중 서로 다른 두 담당자의 담당 부품이 동일 할 수도 있다.

**[입력]**

첫 줄에 테스트 케이스의 개수 T 가 주어지며, 그 다음 T 개의 테스트 케이스가 주어진다.   
각 테스트 케이스는 여러 줄로 구성되어 있으며, 첫 줄에는 각 라인에서 하루에 처리하는 부품의 수 N, 전체 라인의 수 M, 투입되는 엔지니어의 수 K 가 공백으로 구분되어 주어진다.   
그 다음 줄에는 각 엔지니어가 담당하는 부품 알파벳이 K 개 주어진다.   
그 다음부터 M 개의 줄에 걸쳐 각 생산 라인의 일일 생산계획이 주어진다. 각 생산 라인의 일일 생산계획은 N 개의 알파벳으로 주어진다.

**[출력]**

각 테스트 케이스에 대해 #x (x는 테스트 케이스 번호, 1부터 시작)을 출력하고 공백을 하나 둔 다음, 최대 생산 가능한 정상부품의 수를 출력한다.

**[입출력 예]**

(입력)                      
4      // 테스트케이스 개수   
5 5 1  // 1번째 케이스   
A   
AAEFM   
BDEKN   
ADFJN   
ACFIM   
ABGHA   
5 5 1  // 2번째 케이스   
F   
AAEFM   
BDEKN   
ADFJN   
ACFIM   
ABGHA   
5 5 3  // 3번째 케이스   
GFA   
AAEFM   
BDEKN   
ADFJN   
ACFIM   
ABGHA   
5 5 4  // 4번째 케이스   
BAZA   
AACBA   
ABCHU   
ACFIA   
ABKPA   
AAZMA   
  
(출력)                                            
#1 6   
#2 3   
#3 8   
#4 13   
  
(sample\_input.txt 의 출력)   
#1 6   
#2 3   
#3 8   
#4 13   
#5 1   
#6 0   
#7 10   
#8 7   
#9 7   
#10 3   
#11 3   
#12 7   
#13 8   
#14 9   
#15 15   
#16 27   
#17 33   
#18 30   
#19 68   
#20 169