2461: 대표 선수

sianke1991

예시 데이터:
A반 – C반 각 학급에 4명의 학생이 있으며 각 학생의 능력치는 다음과 같다.

A반	12 16 67 43
B반	7 17 68 48
C반	14 15 77 54

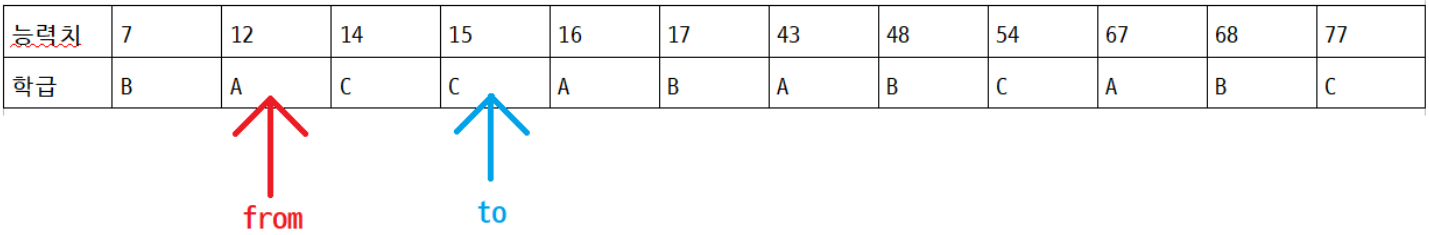
각 반에서 한 명씩 뽑아 시합을 한다. 이 때 선발된 학생들 중 능력치의 최대치와 최소치의 차이가 최소가 되도록 선발하려고 한다. 예를 들어 A반에서 12, B반에서 7, C반에서 77 학생을 선발하면 최대치는 77, 최소치는 7이 되어 그 차이는 70이 된다. 하지만 A반에서 16, B반에서 17, C반에서 15 학생을 선발하면 최대치는 17, 최소치는 15가 되어 그 차이는 2가 된다.

문제에서 입력된 학생 데이터를 그 능력치 기준으로 정렬하면 다음과 같다.

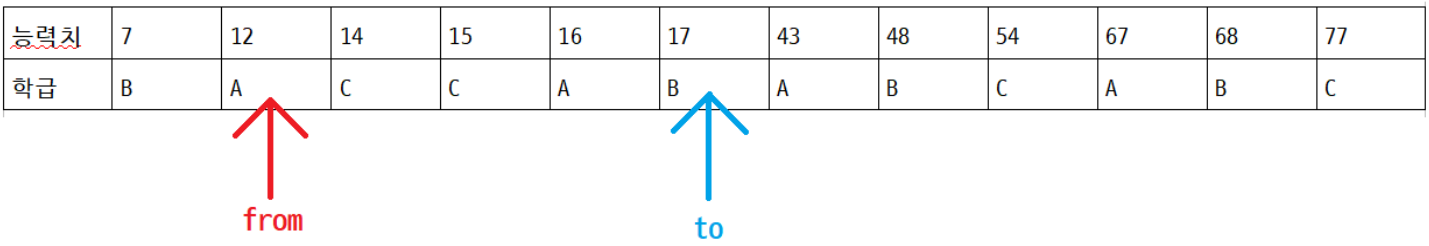
능력치	7	12	14	15	16	17	43	48	54	67	68	77
학급	B	A	C	C	A	B	A	B	C	A	B	C

우리는 이 학생들 중 일부를 선별하는데 능력치의 최대치와 최소치의 차이가 최소가 되기를 원하므로 투 포인터 알고리즘을 적용해 본다. (왼쪽 인덱스에서 능력치의 최소치를, 오른쪽 인덱스에서 능력치의 최대치를 구할 수 있다.)

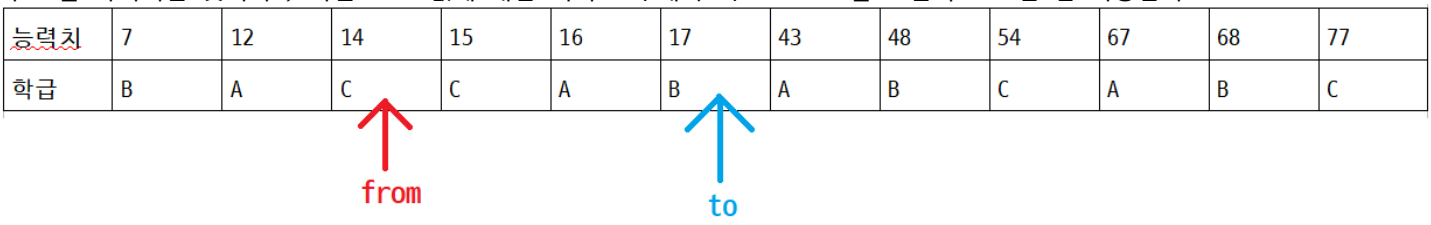
예를 들어 두 인덱스를 아래와 같이 잡으면,



이 영역으로는 B반의 학생이 선별되지 않으므로 to 인덱스를 오른쪽으로 늘려서 최소-



이 지점까지 to 인덱스가 이동해야 한다. 이 지점에서 to 인덱스가 더 오른쪽으로 가도 [각 학급 당 한 명은 뽑는] 조건은 만족되지만, 우리가 구하는 목표인 [최대치와 최소치의 차이가 최소가 되도록 하는 것]에는 크게 도움이 안된다. 즉 from이 12를 가리키고 있다면 to는 17 까지 이동하고 그 이상으로는 이동할 필요가 없다. (from 인덱스가 12를 가리킬 때의 최적은 to 인덱스가 17을 가리키는 것이다.) 다른 from 값에 대한 최적도 구해야 하므로 from을 오른쪽으로 한 칸 이동한다.



from과 to가 위와 같이 가리켜도 [각 학급 당 한 명은 뽑는] 조건은 만족한다. 즉 from이 14를 가리킬 때 최적은 to 인덱스가 17을 가리키는 것이다.
위와 같은 방법으로 from 인덱스가 가리키는 각 값에 대해 to 인덱스의 최적을 구할 수 있다.