1. PPAP

PPAP 문자열의 조건은 다음과 같다.

P는 PPAP 문자열이다.

PPAP 문자열에서 P 하나를 PPAP 로 바꾼 문자열은 PPAP문자열이다.

PPAP 문자열 S를 보자. S가 P가 아니라면 S에는 PPAP 1개 이상 존재한다. 주어진 조건에서 PPAP 문자열에서 P를 PPAP로 바꾼 문자열은 PPAP 문자열이므로 S에 있는 PPAP 를 모두 P로 바꿔도 PPAP 문자열이 된다. 이를 계속 반복하면 S는 P가 된다. 즉 문자열 S에 있는 PPAP를 모두 P로 꾸는 과정을 진행하며 S가 PPAP 문자열이 되지 않는다면 S는 PPAP 문자열이 되지 않는다.

그렇다면 PPAP 패턴을 찾는 방법은 스택을 이용해 문자열을 순회하며 P가 나올 때 마다 스택에 P를 넣고 A가 나온다면 다음 문자가 P이고 스택의 사이즈가 2보다 큰 경우 스택에 있는 P문자를 하나 제거하면 된다.

1. 그렇고 그런 사이

N명의 사람을 일렬로 세웠을 때, i번째 사람의 번호를 ai라 하자. K개의 그렇고 그런 사이를 만들기 위해 어떻게 해야 할까? 먼저 1번 사람과 그렇고 그런 사이 사람의 수는 1번 사람보다 왼쪽에 있는 사람의 수이다. 2번 사람과 그렇고 그런 사이는 사람의 수는 자신보다 왼쪽에 있는 사람들 중, 1번 사람을 제외한 사람들의 수이다. 여기서 만약 작은 번호의 사람을 가장 오른쪽에 우선 배치한다면 자기보다 작은 사람의 수를 구할 필요가 없다.

따라서 만들어야 할 그렇고 그런 사이가 Kr이 남은 배치 가능한 공간 이상이라면 배치 가능한 공간에 가장 오른쪽에 배치 가능한 사람 중, 가장 작은 번호를 배치한다. 만약 그 보다 작다면 Kr+1번째 위치에 배치 가능한 번호 중, 가장 작은 번호를 배치한다. 그리고 남은 공간엔 남은 배치 가능한 사람을 오름차순으로 배치한다면, 앞서 배치한 사람들로 K개의 그렇고 그런 사이가 만들어짐이 보장되고 후에 남은 공간에 배치된 사람들은 서로 그렇고 그런 사이가 되지 않게 된다.

1. 난로

먼저 성냥이 1개밖에 없을 때, 처음부터 끝까지 난로를 켜야 하므로 마지막에 오는 친구가 나가는 시각에서 처음 오는 친구의 도착시간을 빼야 한다. 성냥이 2개가 있다면 처음 킬 때 1개의 성냥을 쓰고 다음 친구가 올 때까지 가장 긴 시간을 보내야 하는 순간에 난로를 끄고 다시 키면 된다. 3개라면 가장 긴 시간을 보내는 순간 2번을 껐다 키면 된다. 즉, K개의 성냥을 가지고 있다면 처음 난로를 키는 순간을 제외하고 K-1번 난로를 껐다 키면 된다. 즉, 먼저 온 친구가 나가고 다음 친구가 들어오는 사이의 간격 N-1개 중에 가장 큰 K-1개의 간격을 고르고 그 간격사이에 난로를 끄면 된다.

1. 정육점

최소한의 비용으로 M이상의 고기를 구매해야 한다. 우리가 먼저 사야할 고기는 무엇일까? 가장 싸면서 덩어리 무게가 많은 고기부터 사면 된다. 중요한 것은 특정 고기를 사면 그 고기보다 싼 고기는 무료라는 점이다. 따라서 싼 가격의 고기부터, 같은 가격이라면 덩어리 무게가 많이 나가는 고기부터 산다면 같은 가격으로 고기를 산다고 치면 같은 가격으로 고기들을 산다면 최소한의 덩어리 개수로 고기를 살 수 있다. 그러나, 이런 알고리즘을 M 이상의 고기를 사는 순간 끝낸다면 최적해가 나오지 않는데 예외 케이스로 고기 덩이가 4개 있고 3그램이상 사야할 때, 덩어리의 가격과 무게들이 [[1,1],[1,1],[1,1],[2,1]] 이라면 1그램 고기 덩어리 3개를 사면 3원이 들지만 2그램 고기 덩어리를 사면 2원이 들고 1그램 고기들을 무료로 살 수 있고 총 4그램 살 수 있다. 따라서 고기들을 우선순위에 맞춰 사가며 전보다 비싼 가격의 고기를 산다면 필요한 돈을 비싼 가격으로 갱신, 같은 가격이라면 해당 고기의 가격을 더하며 현재까지 구매한 고기 덩어리가 M이상인지 확인하며 최소 비용을 갱신하면 된다.

1. 수열의 점수

임의의 수 에 대하여 라면 이다. (단, a,b,c,d 모두 같은 부호를 가지고 있어야 한다.) 따라서 양수들과 음수로 나누어 절댓값이 큰 수를 두 개씩 묶어 곱하고 더한다면 가장 큰 점수를 구할 수 있다. 그러나, 만약 묶은 두 수 중, 하나라도 1인 경우엔 두 수를 곱하는 것이 아닌 두 수를 곱하는 것이 아닌 따로 더하는 것이 더 높은 점수를 얻는다. 또한, 음수들의 개수가 홀수라면 짝을 이루지 못한 수가 하나 존재한다. 만약 이 때, 수열에 0이 존재한다면 그 수와 묶으면 음수 값을 점수에 추가하지 않을 수 있다. 따라서 총 정리하면 수열에 등장한 1의 개수, 양수들로 만들 수 있는 최대 점수, 음수와 0들로 만들 수 있는 최대 점수를 모두 더한다면 수열의 최대 점수를 구할 수 있다.