

<https://www.acmicpc.net/problem/2499>



논이 있을 때 형제가 수확량의 차이를 최소화하여 논을 나눠가질 수 있는 방법을 구하는 문제입니다.

풀이

DP 문제입니다.

$DP[열][동생\ 논\ 의\ 행][동생\ 수확\ 량\ 의\ 합] = 수확\ 량\ 차이\ 의\ 최\ 소\ 값$

전에 동생 논 의 행을 저장하여 조건에 맞게 논이 구성되도록 합니다.

$DP[now][here][x] = \min(DP[now + 1][0][x + 수확량] \sim DP[now + 1][here][x + 수확량])$
을 구하면 됩니다.

수확량을 매번 구할 필요 없이 sum에 연속합을 저장하겠습니다.

저 위의 식을 재귀 함수로 짜게 되면 DP 테이블을 완성할 수 있습니다.

이제 완성된 DP 테이블로 역추적을 해야 합니다.

열을 한 번씩 보면서 높이를 출력해야 합니다.

다음 행의 DP에 저장된 값을 비교하여 최솟값일 때 그 높이가 정답입니다.

다음 행의 논 높이로 0~N까지 비교해가면서 수확량을 계산하여 최솟값을 구해주고 높이를 출력한 뒤 다음 행의 논을 계속 확인해 주면 됩니다.

DP 테이블을 DP[25][40005]로 잡아서 푸신 분들도 있는데 저는 당장 떠올랐던 게 이 풀이라서...

더 빠르게 풀고 싶으신 분들은 도전해보세요!

소스 코드

```

#include <bits/stdc++.h>
#define FIO ios::sync_with_stdio(0), cin.tie(0), cout.tie(0)
#define FOR(i, a, b) for(int i=a;i<=b;i++)
using namespace std;

int main() {
    FIO;
    int n;
    cin >> n;
    int sum[25][25] = {}, total = 0;
    FOR(i, 1, n) FOR(j, 1, n) {
        int x;
        cin >> x;
        sum[i][j] = sum[i - 1][j] + x;
        total += x;
    }
    int dp[25][25][2005];
    memset(dp, -1, sizeof(dp));
    function<int(int, int, int)> DP = [&](int now, int here, int x) {
        int& ret = dp[now][here][x];
        if (ret != -1) return ret;
        if (now > n) return ret = abs(total - x - x);
        ret = 1e9;
        FOR(i, 0, here)
            ret = min(ret, DP(now + 1, i, x + sum[i][now]));
        return ret;
    };
    cout << DP(1, n, 0) << '\n';
    auto path = [&]() {
        int cost = 0;
        FOR(i, 1, n) {
            int Min = 1e9, idx;
            FOR(j, 0, n) {
                int x = cost + sum[j][i];
                if (dp[i + 1][j][x] == -1) continue;
                if (Min > dp[i + 1][j][x]) {
                    Min = dp[i + 1][j][x];
                    idx = j;
                }
            }
            cost += sum[idx][i];
            cout << n - idx << ' ';
        }
    };
    path();
    return 0;
}

```