

Задача 2. Лабиринт

Автор: Тодор Петров

Тази задача е най-стандартната задача за динамично оптимизиране, която се дава във всички учебници. Идеята е съвсем проста: ако искаме да намерим най-добрия резултат, който може да се постигне от клетка (1, 1) до клетка (i, j), то трябва да сметнем най-добрите резултати от (1, 1) до всички клетки (i - 1, k), където k е от (1 до i - 1). По този начин:

$best(i, j) = \max(best(i - 1, j - 1), best(i - 1, j)) + value(i, j)$ {където, $best(i, j)$ е най-добрия резултат от (1, 1) до (i, j), а $value(i, j)$ е стойността на картата в клетка (i, j)}.

Очевидно е, че $best(1, 1) = value(1, 1) \Rightarrow$ последователно можем да сметнем $best(2, k)$ (за всички допустими k), $best(3, k)$ (за всички допустими k) и т.н.

Крайният резултат е $\max(best(n, k))$ за всички допустими k т.е. започвайки от върха на лабиринта и свършвайки някъде по основата му.