

64. Minimum Path Sum

Given a $m \times n$ grid filled with non-negative numbers, find a path from top left to bottom right which *minimizes* the sum of all numbers along its path.

Note: You can only move either down or right at any point in time.

Example:

```
Input:
[
  [1,3,1],
  [1,5,1],
  [4,2,1]
]
Output: 7
Explanation: Because the path 1→3→1→1→1 minimizes the sum.
```

动态规划。矩阵中的某一点由左边和上边的点移动过来的。假设 $p_{i,j}$ 是矩阵左上角到点 $p_{i,j}$ 的最短距离，那么状态方程为： $p_{i,j} = \min(p_{i-1,j} + \text{grid}_{i,j}, p_{i,j-1} + \text{grid}_{i,j})$

```
class Solution {
public:
    int minPathSum(vector<vector<int>>& grid) {
        int row = grid.size(), col = grid[0].size();
        //动态申请二维数组
        int **p = new int*[row];
        for(int i=0;i<row;i++){
            p[i] = new int[col];
        }
        p[0][0] = grid[0][0];
        for(int i=1;i<row;i++){
            p[i][0] = grid[i][0] + p[i-1][0];
        }
        for(int j=1;j<col;j++){
            p[0][j] = grid[0][j] + p[0][j-1];
        }
        for(int i=1; i<row;i++){
            for(int j=1;j<col;j++){
                p[i][j] = min(grid[i][j]+p[i-1][j], grid[i][j]+p[i][j-1]);
            }
        }
        return p[row-1][col-1];
    }
};
```