

64. Minimum Path Sum

题目描述: <https://leetcode.com/problems/minimum-path-sum/>

给定一个二维数组，求从左上走到右下的最短距离。

解题思路:

DP

代码 v.1.0

```
class Solution {
public:
    int minPathSum(vector<vector<int>>& grid) {
        int m = grid.size();
        int n = grid[0].size();
        vector<vector<int>> f(m, vector<int>(n,0));
        f[0][0] = grid[0][0];
        for(int j = 1; j < n; j++){
            f[0][j] = grid[0][j] + f[0][j-1];
        }
        for(int i = 1; i < m; i++){
            f[i][0] = grid[i][0] + f[i-1][0];
        }
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                cout << f[i][j]<<" ";
            }
            cout <<endl;
        }
        for(int i = 1; i < m ; i++){
            for(int j = 1; j < n; j++){
                f[i][j] = min(f[i][j-1], f[i-1][j]) + grid[i][j];
            }
        }
        return f[m-1][n-1];
    }
};
```

代码 v.2.0

```

class Solution {
public:
    int minPathSum(vector<vector<int>>& grid) {
        int m = grid.size();
        int n = grid[0].size();
        vector<int> f(n+1,grid[0][0]);
        for(int i = 1; i < n; i++){
            f[i+1] = f[i] + grid[0][i];
        }
        f[0] = INT_MAX-3;
        for(int i = 0; i <= n; i++)
            cout<<f[i]<<" ";
        cout<<endl;
        for(int i = 1; i < m ; i++){
            for(int j = 0; j < n; j++){
                f[j+1] = min(f[j],f[j+1]) + grid[i][j];
            }
        }
        return f[n];
    }
};

```