120. Triangle

Description | Hints | Submissions | Discuss | Solution

Pick One

Given a triangle, find the minimum path sum from top to bottom. Each step you may move to adjacent numbers on the row below.

For example, given the following triangle

```
[
     [2],
    [3,4],
   [6,5,7],
  [4,1,8,3]
]
```

The minimum path sum from top to bottom is `11` (i.e., **2** + **3** + **5** + **1** = 11).

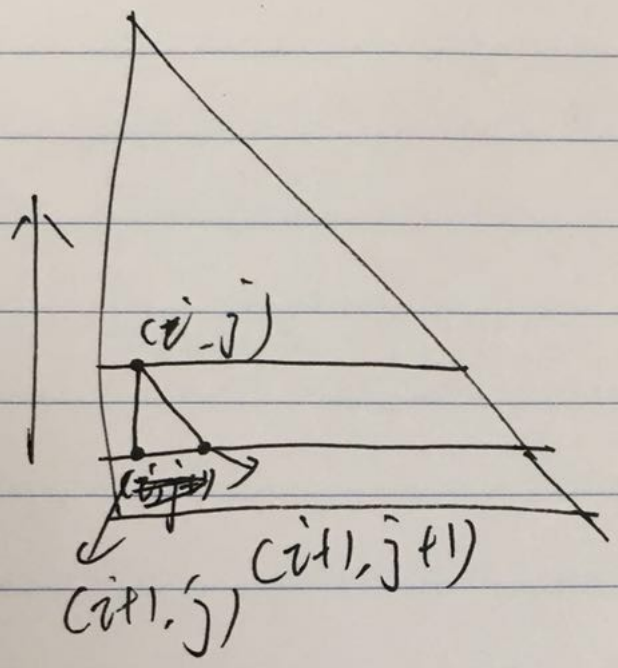
```java
/*
 * 这道题目是典型的动态规划题目
 * 递推公式是：res[i][j]=Math.min(res[i-1][j-1],res[i-1][j])+list(i).get(j)
 * 即从上一行能够到达该点的两个点钟选择小的那个点，然后到达该点加上当前点的值即可
 * 这道题目由于是个三角形，无需建立二维数组。
 */
public class L120 {
    public int minimumTotal(List<List<Integer>> triangle) {
        if(triangle.size()==1)
            return triangle.get(0).get(0);

        int[] dp = new int[triangle.size()];

       //题目从下到上，首先dp[i]设置为最后一行的值
        for (int i = 0; i < triangle.get(triangle.size() - 1).size(); i++) {
            dp[i] = triangle.get(triangle.size() - 1).get(i);
        }

        // 然后从倒数第二行开始，递推上去
        for (int i = triangle.size() - 2; i >= 0; i--) {
            for (int j = 0; j < triangle.get(i).size(); j++) {
                dp[j] = Math.min(dp[j], dp[j + 1]) + triangle.get(i).get(j);
            }
        }

        return dp[0];
    }
}
```



(i,j) dp[j] = min(dp[i+1,j], dp[i+1,j+1]) + d_triangle(i,j).

这样得到的是(i,j)的最小值. (从底往