Triangle

 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).

首先，想到的是从上往下解决这个问题。设每个点的值为， 每个点的最小路径为.

于是我们可以有

从中我们可以推出我们的转移方程：

于是可以写出代码 （TtoB.cpp）。但是这种从上往下的方法会把每个中间状态都缓存起来，这样的话，我们的空间复杂度就有。 所以可以有更好的方式来解决空间问题。

后来我看到了从下往上的解决方法，思路与从上往下很相似，不过不用把每个点的状态记录下来。每次最下层的状态用完后就可以“抛弃”，因为再往上走的时候不需要用它们了，而不像从上往下的状况，每次都需要用到最上层的缓存记录。同样的我们可以得到转移方程：

(当前的状态可以有下一层相邻的两个状态得到)

同时，对于时的状态，我们不需要再保留，因此上面的公式可以进一步简化为

最后只需要取就是我们需要值。 这样整个运算的空间为

(BtoT.cpp)