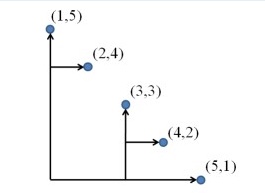
# I - 最小生成树？

##### Time Limit: 3000/1000MS (Java/Others)     Memory Limit: 65535/65535KB (Java/Others)

Submit Status

坐标平面上有nn个点，现在我们想要把这些点连成一棵有向树。但是不幸的是，树的边只能向右或者向上长。下面是一张示意图。



现在我们想知道，这棵树总边长的最小值是多少。

## Input

有多组测试数据，对于每一组数据:

第一行有一个数，nn

接下来nn行，每一行有两个数。第ii行的两个数分别是xixi和jiji。

0<n≤1030<n≤103

0≤xi,yi≤1040≤xi,yi≤104

数据保证对于所有的xi≤xjxi≤xj，有yi≥yjyi≥yj。

## Output

对于每一组数据，输出一个整数，表示最小的，树的总边长

## Sample input and output

| **Sample Input** | **Sample Output** |
| --- | --- |
| 5  1 5  2 4  3 3  4 2  5 1  1  10000 0 | 12  0 |

## Hint

最小生成树?

题意：

将n个坐标点用横竖两个方向的线连接，使线的总长最小。

题解：

区间DP+四边形不等式优化。

首先，我们可以通过观察得知，连接坐标为(x1,y1)和(x2,y2)两点的代价(长度)就是abs(y1-y2)+abs(x1-x2)。此时，我们可以用横纵方向的两条新线所得到的交点来代替原有的两点。这时，我们可以很容易的得到递推方程：

dp[i][j]=min(dp[i][k]+dp[k+1][j]+abs(x[k+1]-x[i])+abs(y[k]-y[j]))

(i<=k<j)

此时，算法的时间复杂度约为O(n^3),而n最大可为10^3,很明显这时程序几乎铁定会超时。

我们该如何优化算法呢？这时，我们想到了一道与本题十分相似的题目——石子合并。于是，我们可以用类似的方法证出dp[i][j]满足四边形不等式，并在此基础上进一步用相同的方法优化：

dp[i][j]=min(dp[i][k]+dp[k+1][j]+abs(x[k+1]-x[i])+abs(y[k]-y[j]))

(s[i][j-1]<=k<=s[i+1][j])

其中，s[i][j]表示合并编号i~j之间的点时，代价最小时的最后一个合并点。如：s[i][j]=k，表示合并编号i~j之间的点时最后一步合并i~k与k+1~j这两个区间能取得最优解。

这样，优化后的算法便有了O(n^2)的时间复杂度，符合题意。