# 1.DSA构思

## 思路来源声明

一开始几乎没有思路，参考了farui、 Moonshile和jacobkusch提供的思路：

<http://www.xuetangx.com/courses/TsinghuaX/30240184X/2014_T2/discussion/forum/i4x-edx-templates-course-Empty/threads/54524c4a459f08a4dc00d31e>

### 通项公式推导

有思路后，开始推导Edi\*wi的通项公式。

E0 = x1w1 + x2w2 + x3w3 + x4w4 + x5w5 + ... + xnwn

E1 = (x2 - x1)w2 + (x3 - x1)w3 + (x4 - x1)w4 + (x5 - x1)w5 + (x6 - x1)w6 + ... + (xn - x1)wn

E2 = (x2 - x1)w1 + (x3 - x2)w3 + (x4 - x2)w4 + (x5 - x2)w5 + (x6 - x2)w6 + ... + (xn - x2)wn

E3 = (x3 - x1)w1 + (x3 - x2)w2 + (x4 - x3)w4 + (x5 - x3)w5 + (x6 - x3)w6 + ... + (xn - x3)wn

E4 = (x4 - x1)w1 + (x4 - x2)w2 + (x4 - x3)w3 + (x5 - x4)w5 + (x6 - x4)w6 + ... + (xn - x4)wn

E1 - E0 = -x1w1 - x1w2 - x1w3 - x1w4 - x1w5 - x1w6 - ... - xnwn

= -x1Ewn

= 2(x1 - x0)w0 + (x0 - x1)Ewn

E2 - E1 = (x2 - x1)w1 + (x1 - x2)w2 + (x1 - x2)w3 + ... + (x1 - x2)wn

= 2(x2 - x1)w1 + (x1 - x2)Ewn

E3 - E2 = (x3 - x2)w1 + (x3 - x2)w2 + (x2 - x3)w3 + (x2 - x3)w4 + ... + (x2 - x3)wn

= 2(x3 - x2)(w1 + w2) + (x2 - x3)Ewn

E4 - E3 = (x4 - x3)w1 + (x4 - x3)w2 + (x4 - x3)w3 + (x3 - x4)w4 + ... + (x3 - x4)wn

= 2(x4 - x3)(w1 + w2 + w3) + (x3 - x4)Ewn

// 特殊的，若第一个村里坐标不为0，则其Edw为E0 – x\*EWN.

// EWN为所有村落的权重之和。

整理可得：

设总村落数量为n，对于k∈[0,n]

Ek – Ek-1 = 2(Xk – Xk-1) + (Xk – Xk-1)

从而：

**Ek = Ek-1 + 2(Xk – Xk-1) + (Xk – Xk-1)**

## 第一次实现

以结构体保存xi与wi，建立结构体数组。使用快速排序算法，将数组按照xi排序。

计算第一个村落的结果后，设其为最小值，开始1~n的循环，若发现有比最小值更小的，更新最小值。

但是超时了。分析应是排序拖慢了运算速度。那么就开始考虑不排序的算法。

重新读题目发现，坐标限制为[0,32767]，则可以使用32768大小的数组存放村落权值，其下标就是其坐标，从而节省了空间。更重要的是，可以在输入时就将村落按照坐标排序了！输入完它就是排好序的！太棒了！

## 第二次实现

以数组存放村落权值，其下标就是其坐标。

可以干活了！

# 2.问题与解决方法

合并相同坐标的村落时，我只保留了最大的权值，但是没有发觉，两个村落虽然坐标相同被合并，但是其权值不能简单的二选一保留大值，而应该将二者相加！这样才符合现实啊。

# 3.时间复杂度与空间复杂度

时间：输入敏感，O(n)

空间：O(n)

# 4. 介绍理论分析与实测效果的吻合程度，不吻合时进一步解释原因

略

# 5. 所用方法的特别、新颖或创新之处

略

# 6.待优化思路

题目约定最多32768个村落，但有可能不全，并稀疏分布。若遍历整个数组判断其是否有效则效率有点低。下一步想使用之前学到的Bitmap结构保存有效村落下标，以加快扫描速度。