Lesson2

排序不等式

排队打水

思路：

- 总共等待时间为：t1 * (n-1) + t2 * (n-2) + t3 * (n-3) + ... + t(n-1) * 1，且乘的系数n-1 > n-2 > ... > 1
- 故**从小到大排序**，总时间最小

证明：（反证法）

- 假设最优解不是从小到大排序，则必然存在两个相邻元素，前一个ti比后一个t(i+1)大，此时我们交换ti和t(i+1)的位置。交换前等待ti和t(i+1)的时间为ti * (n-i) + t(i+1)(n-i-1)，而交换后的等待时间为t(i+1) * (n-i) + ti * (n-i-1), 且ti * (n-i) + t(i+1)(n-i-1) - [t(i+1) * (n-i) + ti * (n-i-1)] = ti - t(i+1) > 0，所以交换ti与t(i+1)之后，总时间将严格变小。与其是最优解矛盾，因此最优解一定是从小到大排序。

具体实现

```
static int N = 100010;

static int n;
static int[] t = new int[N];

public static void main(String[] args) throws Exception {
    ins.nextToken(); n = (int)ins.nval;

    for (int i=0; i<n; i++) { ins.nextToken(); t[i] = (int)ins.nval; }

    Arrays.sort(t, 0, n);   //将t[i]从小到大排序

    long res = 0;
    for (int i=0; i<n; i++) res += t[i]*(n-i-1);

    out.println(res);

    out.flush();
}
```

绝对值不等式

货舱选址

思路

- f(x) = | x1 - x | + | x2 - x | + ... + | xn - x |
- 猜测：若xn为奇数，则当x取到中位数时函数值最小；若xn为偶数，则当x取到中间两个数之间（包括左右两个端点）时最小

证明：

- 将右侧分成若干组，变为

  f(x) = (| x1 - x | + | xn - x |) + (| x2 - x | + | x(n-1) - x |) + ...

  考虑|a-x|+ |b-x|，当x落在[a, b]之间时 |a-x|+ |b-x| 取最小值 |b-a|

  所以上述f(x)等式右侧 >= (xn - x1) + (x(n1-)-1) + ...

  对于没一项，取等的条件分别为 x ∈ [x1, xn]，x ∈ [x2, x(n-1)]，...

  所以若xn为奇数，则当x取到中位数时函数值最小；若xn为偶数，则当x取到中间两个数之间（包括左右两个端点）时最小。

具体实现

```
static int N = 100010;

static int n;
static int[] a = new int[N];

public static void main(String[] args) throws Exception {
    ins.nextToken(); n = (int)ins.nval;

    for (int i=0; i<n; i++) { ins.nextToken(); a[i] = (int)ins.nval; }

```

```
    Arrays.sort(a, 0, n);   //从小到大排序，用于求中位数

    long res = 0;
    for (int i=0; i<n; i++) res += Math.abs(a[i]-a[(n-1)/2]);

    out.println(res);

    out.flush();
}
```

推公式

耍杂技的牛

思路：

- 按照wi+si从小到大的顺序排，最大的风险系数一定是最小的。

证明：(ans >= cnt && ans <= cnt)

- 该贪心得到的答案，一定大于等于最优解(ans >= cnt)
- （反证法）若不是按照wi+si从小到大的顺序排，则一定存在相邻两头牛，且**wi+si > w(i+1)+s(i+1)**，交换第i个位置与第i+1个位置上的牛。交换前，第i个位置牛的危险系数为: w1+w2+...+w(i-1)-si， 第i+1个位置牛危险系数为: w1+w2+...+wi-s(i+1)。交换之后，第i个位置牛的危险系数为: w1+w2+...+w(i-1)-s(i+1)，第i+1个位置牛危险系数为: w1+w2+...+w(i+1)-si。每个位置危险系数同时减去w1+w2+...+w(i-1)，加上si与s(i+1)，所以现在每个位置的危险系数为**s(i+1)[交换前i]，wi+si[交换前i+1]，si[交换后i]，w(i+1)+s(i+1)[交换后i+1]**。且交换后si[交换后i]，w(i+1)+s(i+1)[交换后i+1]严格小于wi+si[交换前i+1]，因此也严格小于max(s(i+1)[交换前i]，wi+si[交换前i+1])。所以交换之后，其余所有的风险系数不变，而交换之后的两个风险系数变小。重复这一操作，序列中最大值一定不会变大，只可能变小或者不变，因此贪心得到的答案一定小于等于最优解，即ans <= cnt
- 综上所述，ans = cnt

具体实现

```
static int N = 500010;

static int n;
static Cow[] cows = new Cow[N];

public static void main(String[] args) throws Exception {
    ins.nextToken(); n = (int)ins.nval;

    for (int i=0; i<n; i++) {
        ins.nextToken(); int w = (int)ins.nval;
        ins.nextToken(); int s = (int)ins.nval;
        cows[i] = new Cow(w+s, w);
    }

    //按w+s从小到大排序
    Arrays.sort(cows, 0, n, (o1, o2) -> (o1.l-o2.l));

    long res = (int)-2e9, sum = 0;
    for (int i=0; i<n; i++) {
        int s = cows[i].l-cows[i].r, w = cows[i].r;
        res = Math.max(res, sum-s);
        sum += w;
    }

    out.println(res);

    out.flush();
}

static class Cow {
    int l, r;
    Cow(int ll, int rr) {
        l = ll; r = rr;
    }
}
```