以牛客网这道题为例当做模板题：

链接：[https://www.nowcoder.com/acm/contest/93/I](https://www.nowcoder.com/acm/contest/93/I" \t "_blank)  
来源：牛客网

wyh学长现在手里有n个物品，这n个物品的重量和价值都告诉你，然后现在让你从中选取k个，问你在所有可能选取的方案中，最大的单位价值为多少（单位价值为选取的k个物品的总价值和总重量的比值）

## 输入描述:

输入第一行一个整数T(1<=T<=10)  
接下来有T组测试数据，对于每组测试数据，第一行输入两个数n和k(1<=k<=n<=100000)  
接下来有n行，每行两个是a和b，代表这个物品的重量和价值

## 输出描述:

对于每组测试数据，输出对应答案，结果保留两位小数

链接：[https://www.nowcoder.com/acm/contest/93/I](https://www.nowcoder.com/acm/contest/93/I" \t "_blank)  
来源：牛客网

1

3 2

2 2

5 3

2 1

## 输出

0.75

## 说明

对于样例来说，我们选择第一个物品和第三个物品，达到最优目的

这是一道最经典的01分数规划，看似像是背包问题，其实不然，这类问题也是有n个物品每个物品有价值和重量，选取K个，求总价值和重量比值怎么样

和背包不同的是，固定了你必须选取多少个，但是不去限定你的背包最多选多少重量的物品。

这类问题解法是自成一体的,,完全不是背包问题

设v=f(x)=a(x)/b(x);

Val[i]是第i个物品价值，w[i]是第i个物品重量

v=f(x)是最值。a(x)是选取x个总价值，b(x)是选取x个的总重量   
倒着推：存在 v\*=f(x∗)v∗=f(x∗) 为该规划的最优解，变形后:

0=a(x∗)−v∗⋅b(x∗)0=a(x∗)−v∗⋅b(x∗)

我们构造g(v)g(v) 函数：

以最小为例

g(v)=min{a(x)−v⋅b(x)} 其中x属于[0,n],v是个浮点数

* g(v)是关于v的函数,当v是最值v\*的时候a(x)−v⋅b(x)=0=g(v\*)；

**上边那句话就是：Dinkelbach定理叙述为：**设λ∗λ∗为原规划的最优解，则g(λ)=0g(λ)=0当且仅当λ=λ∗λ=λ∗。

把v固定，x从0到n变化，a(x)−v⋅b(x)取最小的

若v=v+1; x从0到n变化，a(x)−v⋅b(x)取最小的

取最小的情况下，也是min(a(x)−v⋅b(x))> min(a(x)−(v+1)⋅b(x))

* 上面可以看出来，g(v)g(λ)是一个严格递减函数

如何计算答案：

g(v)=0时候，v一定是在从n个物品选取x个的正确答案，g(v)=min{a(x)−v⋅b(x)}，x=k时候，g(v)=min{a(k)−v⋅b(k)}

a(k)是k个val[]之和，新的数组new\_val设每个是b(k)是k个w[]之和,新的数组new\_w

把他们拆出来

g(v)=new\_val[0]-v\*new\_w[0] + new\_val[1]-v\*new\_w[1]

+...+ new\_val[k-1]-v\*new\_w[k-1]=d0+d1+d2+...+d[k-1]

d[i]数组代表第i个数的val[i]-v\*w[i]

最优解是g(v)==0时那么就要需寻找k个d相加得0

怎么找这k个物品。

那就要看v了，要用到前面的性质g(v)g(λ)是一个严格递减函数

用二分的去试验v,然后对于每个v，求出n个物品的d

d[0]=val[0]-v\*w[0]

d[1]=val[1]-v\*w[1]

.......

d[n-1]=val[n-1]-v\*w[n-1]

把它们排序，找出前k个看看是不是等于0

等于0说明找对了v

否则说明v过大过小，向哪个如果方向看题，

如果你要求a(x)/b(x)最大，v试验发现，g(v)是正数，v下次就要往大了取

## 时间复杂度：

二分是logn,排序是n\*logn;总是n\*(logn)^2