2025/11/29 18:34 1/2 WQS二分

WQS二分

 wqs二分可以解决形如在\$n\$个物品中选\$k\$个物品最大化/最小化总价值。直接dp一般是\$O(nk)\$的, 而如果题目满足下面的条件并且可以用\$O(n)\$时间在不考虑选择物品个数的情况下计算出最大值/最 小值,就可以用wqs二分做到\$O(n \log n)\$□

- 以最大化价值为例,设\$x\$表示选择物品的个数,设\$f(x)\$表示选\$x\$个物品的最大值,如果\$f(x)\$关于\$x\$是凸函数或凹函数,那么图象就是一个凸包。虽然我们不知道具体凸包长什么样,但是我们可以利用凸包的性质,通过二分直线的斜率使得这条直线与\$(k,f(k))\$这个点"相切",从而间接得到\$f(k)\$的值。
- 以凸函数为例,由于图像是一个凸包,所以一条直线,如果想让直线与图象有交点的情况下截距最大,那么直线一定恰好经过凸包上的一个点,此时我们通过下面的方法求出这个点后,如果这个点的横坐标\$<k\$□那么减少斜率,否则增大斜率。
- 设直线为斜率为\$mid\$□直线方程即为\$y = mid \times x + b\$那么截距为\$y mid \times x\$□当直线 经过每个点时截距为\$f(x) mid \times x\$□设经过\$z\$时截距最大,那么\$\max\{f(x) mid \times x\] = f(z) mid \times z\$□也就是说如果给每个物品的价值减少\$mid\$□那么最大化价值时选择的物品数就是\$z\$□此时不用考虑需要选择的物品数,只需要记录每个状态选择了几个物品即可,一般通过一次dp即可获得结果。假设最后得到的斜率为\$I\$□此时最大截距为\$b\$□那么答案即为\$b + I * k\$□
- 需要注意,一般题目中的\$f(x)\$可能并不是严格凸或者严格凹,因此可能出现连续的点对应斜率相同,二分的时候要注意边界问题。
- 可以开一个结构体记录dp状态的最大值以及取得此时最大值的选取物品数,大幅简化代码编写的复杂度。

CF739E

- 题意:抓宝可梦,有\$n\$个宝可梦和\$a\$个普通球和\$b\$个超级球,每种球抓到每个宝可梦的概率不同, 对一个宝可梦可以不用球、只用普通球或超级球、两种球都用(抓到只算一个),求抓到宝可梦数 量期望的最大值。
- 题解:可以发现,只考虑某一种球的情况下,用\$x\$个该球能抓到宝可梦数量期望是上凸的,因此可以wqs二分套wqs二分。外层二分普通球的斜率,内层二分超级球的斜率。

林克卡特树

- 题意:给出一颗\$n\$个节点的树,每条边有边权,从中挑选\$k+1\$个互不相交的链,使得所有被选中的边的权值和最大。
- 题解:设\$f(x)\$为选择\$x\$个互不相交的链的边权最大值,这个函数也是上凸的,太菜了不会证。然后就可以wqs二分转化为树形dp□每条链附加一个\$-mid\$的权值即可。这个树形dp的过程非常巧妙,通过微调枚举顺序减少了代码的复杂度。设\$f[x][i]\$表示只考虑点\$x\$所在的子树,且点\$x\$的度为\$i\$时的最大权值,因为只能出现链所以\$0 \le i \le 2\$□子树中代码如下,注意枚举顺序不能改,因为前两个转移都用到了上次的结果。

```
void dfs(int i, int fa){
    f[i][0] = Data(0, 0), f[i][1] = f[i][2] = Data(-mid, 1);
    for(int j = 0; j < v[i].size(); j++){
        int x = v[i][j].first, y = v[i][j].second;
        if(x == fa) continue;
        dfs(x, i);
        Data g = max(f[x][0], max(f[x][1], f[x][2]));
        f[i][2] = max(f[i][1] + f[x][1] + Data(y + mid, -1), f[i][2] +</pre>
```

```
update:
2020/06/06
21:51
```

```
g);

f[i][1] = max(f[i][0] + f[x][1] + Data(y, 0), f[i][1] + g);

f[i][0] = max(f[i][0], f[i][0] + g);
}
```

Tree I

- 题意:给出一个\$n\$个点的图,每条边有权值和黑白色两种颜色,求恰好有\$k\$条白边的最小生成树。
- 题解:设\$f(x)\$为选\$x\$个白边的最小生成树,这个函数是下凸的。给每个白边减去\$mid\$跑Kruskal即可。这里优先选白边,那么每次得到的是数个共线点中最靠右的点,二分边界时要注意一下。

CF1279F

• F题

From:

https://wiki.cvbbacm.com/ - CVBB ACM Team

Permanent link:

 $https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:farmer_john:jjleo:wqs\%E4\%BA\%8C\%E5\%88\%86\&rev=1591451460.php.$

Last update: 2020/06/06 21:51



https://wiki.cvbbacm.com/ Printed on 2025/11/29 18:34