动态规划—带权二分优化DP 学习笔记

引入

带权二分其实并不一定用于优化 DP，也可能用于优化贪心等最优化的算法。

带权二分也叫 WQS 二分，最初由王钦石在他的 2012 年国家集训队论文中提出。

定义

使用情况

- 要解决一个最优化问题（求最大 / 最小值）
- 有一个限制，一般是某个参数要求一定恰好为 k

而带权二分就可以很好的解决［**恰好 k 个**］的限制；以选物品取最大收益为例：

- 设 $f(k)$ 为恰好选 k 个时的最大收益，将所有的 $(k,f(k))$ 画出来，图像必须组成一个凸包。
- 因此就可以打表看，是否组成了一个凸包，如果是，则可以考虑带权二分优化。

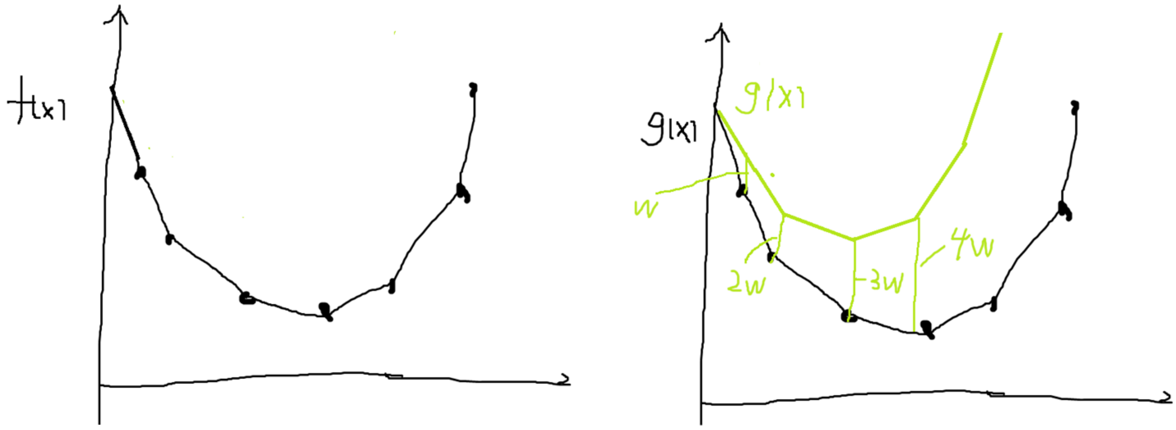
使用方法

例：求 $f(k)$ 的值，我们不会求 $f(k)$ 或者其复杂度不可接受，但是我们可以求出所有 $1\sim n$ 中的最优解 $f(t)$ 及对应的 t，因此我们就可以通过一些处理，将 $f(k)$ 变为最小值，即将全局最优解变为 k。

什么方法？我们设一个额外的 w，每次选一次物品以后就将结果加上 w，也就是，我们设一个新的转移方程 $g_k=f_k+kw$。

> 注意：严谨的，是『选一次』，不是『选一个』。因为有的题目选一次对应一段区间，即多个物品。

可以预见到的，原函数（左）会变为（右）：



要注意的是，w 也可能为负；因此总会有一个 w 使得全局最优解为 $g(k)$，如下：

- 可以想见，如果 w 继续增大，那么最小值点 x_0 会继续变小；
- 如果 w 减小以至于变成负数，那么最小值点 x_0 会不断变大；

那么总会有一个 w，使得最小值在 k 处取到，那不就可以二分了嘛。

我们二分一个值 w（边界可以设置地大一些，当然也可能得根据题目的数据范围调一调），现在问题转化为，求 $g(x)$ 的最小值点 x_0。

此时，不管用 DP 还是贪心啥的方法，求出 $g(x)$ 的最小值 $g(x_0)$，顺便求出此时的最小值点 x_0。

- $x_0 < k$，那就让 w 变小点；
- $x_0 > k$，那就让 w 变大点。

最终，我们就能让 $x_0 = k$，也就是当全局最优解取到 $g(k)$ 的时候，我们是不是就成功的求出了原问题 $f(k) = g(k) - kw$ 呢？

警惕

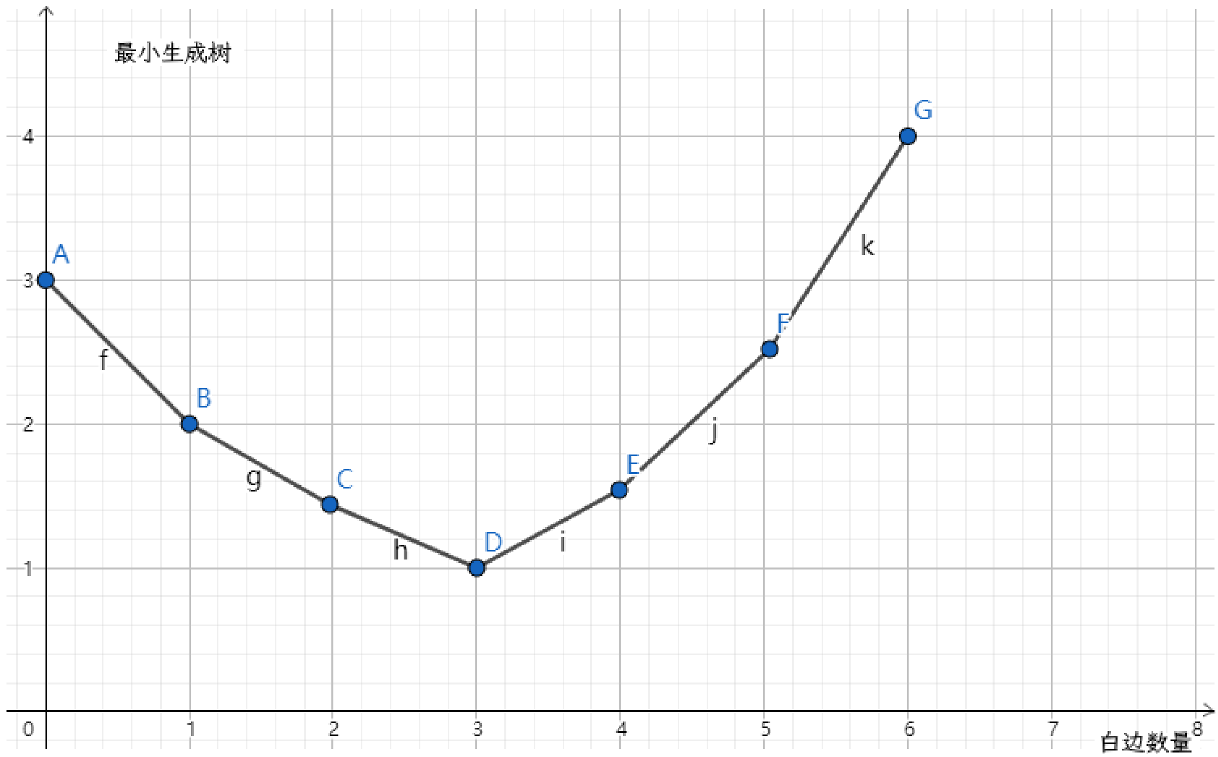
既然是二分，就一定要仔细考虑 l, r, mid 的取值以及更新边界的条件，总之，注意代码细节！

应用

例题讲解

题目：P2619 Tree I

- 画出函数来：



- 凸函数好吧，所以给白边加一个 w 的额外权：

```
int l = -110, r = 110;
while (l <= r) {
    int mid = l + r >> 1; Kruskal(mid);
    if (now0 >= k) {
        ans = ansg - k * mid;
        r = mid + 1;
    } else r = mid - 1;
}
```

- 此题完结。

题单

见：https://www.luogu.com.cn/training/393257

Reference

[1] https://www.mina.moe/archives/6349

[2] https://www.cnblogs.com/Dfkuaid-210/p/wqs_bisect.html

[3] https://blog.csdn.net/Emm_Titan/article/details/124035796

[4] https://blog.csdn.net/weixin_45429627/article/details/109270538

[5] https://blog.csdn.net/Huah_2018/article/details/113796445

本文来自博客园，作者：RainPPR，转载请注明原文链接：https://www.cnblogs.com/RainPPR/p/wqs-binary-dp.html

---

合集：学习笔记

标签：算法 ， 学习笔记