WQS二分简述

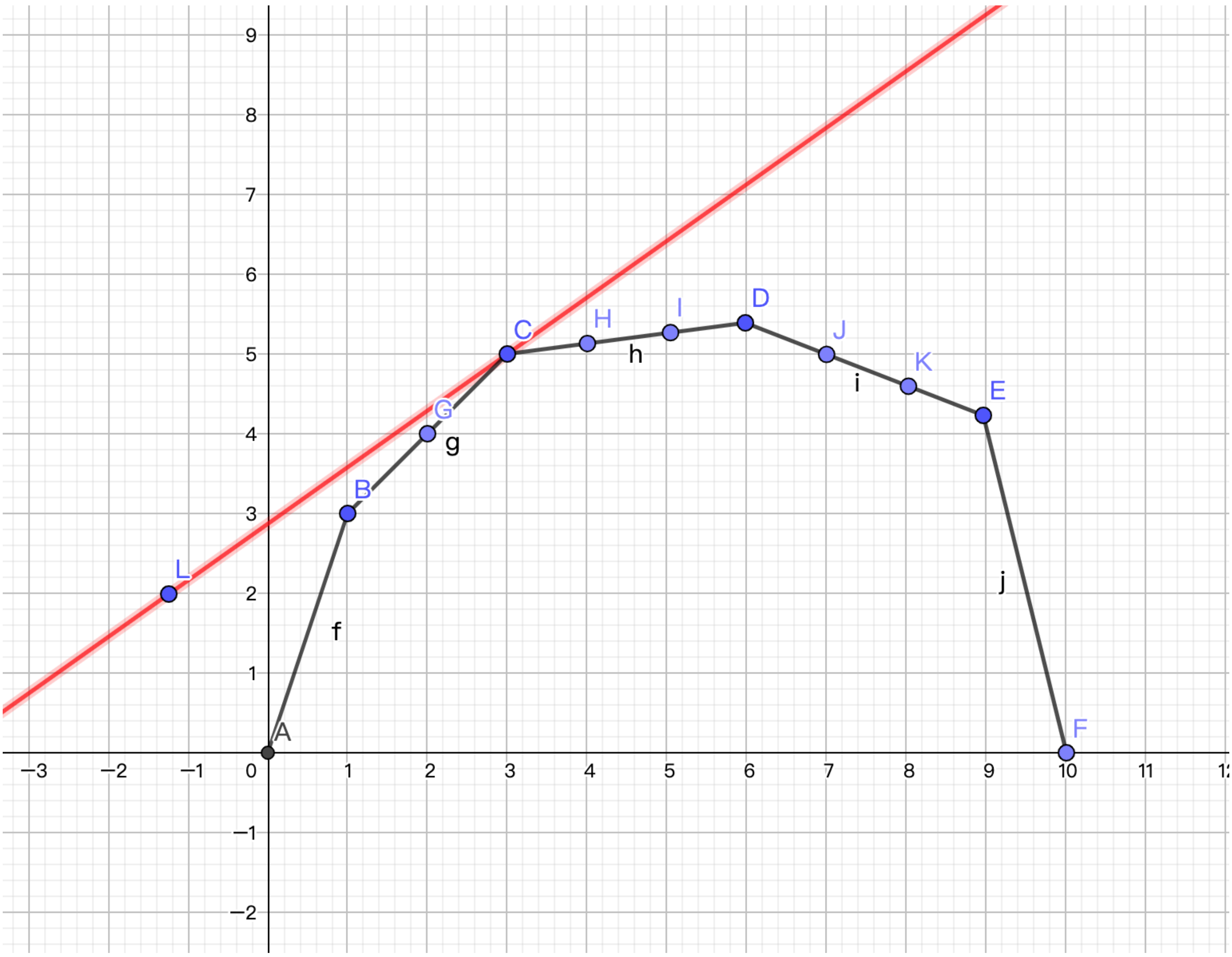
简单的背包问题 & 结论证明

引入

考虑一个问题：一个集合中有 n 个物品，每个物品有一个价值(可能为负)，请你取出 m 个物品，并且使价值之和最大。这种问题一般人选择直接排序，但是如果一个物品的价值计算方式变化了导致不能直接排序取出，甚至没有贪心策略时，应该如何做？如果直接使用二维dp还会超时有什么办法可以快速做？

推结论

实际上这个问题就是WQS问题的典型问题，我们设函数 $\mathrm{g}(x)$ 表示取出 x 个物品时能取到的价值最大值，将每一个 $(x, \mathrm{g}(x))$ ，在一个平面直角坐标系上面画出来，串起来。先忽略红色直线，大概是这个样子，观察可知其凸性：



可以想到的是对于任意一个斜率为 k 的直线都能在上凸壳上找到一个点使得其直线截距最大。

假设现在已知一个斜率为 k 的直线，切到了 $(x, \mathrm{g}(x))$ 这个点，并且截距最大。现在已知 x (接下来默认 x 时定值，用 x 这个记号只是提醒读者有普适性)，如何求解 $\mathrm{g}(x)$ 。

令 $\mathrm{f}(x) = \mathrm{g}(x) - k \times x$ ，也就是每个物品价值减去 k 后能取到 x 个的最大价值。实际上 $\mathrm{f}(x)$ 为在没有拿多少这个限制的时候，所有物品的价值减去 k 后的最优解。

证明如下：由于我们 x 是定制了，换个记号，令 $\mathrm{h}_k(q)$ 表示每个物品减去 k 无限制拿 q 个的最大价值，显然 $\mathrm{h}_k(q) = \mathrm{g}(q) - k \times q$ ，则有 $\mathrm{h}_k(q) + k \times q = \mathrm{g}(q)$，也就是 h 表示用斜率为 k 的直线去切 $(q, \mathrm{g}(q))$ 的截距。

这时我们考虑减去 k 之后有没有可能有更大的取值，也就是当 q 为任意值的时候有没有可能有更大的 $\mathrm{h}_k(q)$ ，设任意的 q 记作 x 。假如说有一个取值更大，那么则存在一个更大的 $\mathrm{g}(x) - k \times x$ ，但是由于刚才的推论 $\mathrm{g}(x) - k \times x$ 表示用斜率为 k 的直线去切 $(q, \mathrm{g}(q))$ 的截距。而 k 固定，凸包上只有一个点可以使得这个东西最大，也就是斜率为 k 的直线切到的点，根据刚才定义这个点就是 $(x, \mathrm{g}(x))$ 。

这个结论使得我们找到了一个可以快速计算的函数 $\mathrm{f}(x)$ ，并且可以转化到 $\mathrm{g}(x)$ 上去，从而实现快速求解。

实现

现在回到做题，我们已知了 x ，并且可以快速计算 $\mathrm{f}(x)$ ， k 未知，目标求解 $\mathrm{g}(x)$ 。实际上我们只需要找到这个 k 就可以了，考虑 k 单调递增变化时，在凸包上切出来的点必定按照 X 递增。那就好办了，直接二分 k 直到切到的点的 X 坐标就是 x 即可。

实际应用中，对于小数二分常常直接限定循环次数；而二分的 `check()` 函数就是跑一遍无限制dp只不过转移方程需要减去一个 mid (二分中的 mid)，然后记录转移次数，并于 x 比较，根据实际情况去二分左区间还是右区间(上凸壳/下凸壳之分)。

推广

实际上对于其他有限制选定个数的dp题目，都可以用上WQS二分，只需要保证 ~~(打表观察)~~ 其有凸性就可以WQS二分。