T1 solution

只用最小面额一定最优，暴力枚举用了多少 C 国币即可。

T2 solution

首先考虑最优，显然每次取能走的里编号最小的，一定能取到最优的答案。这部分可以用小根堆维护。

再来考虑最劣，我们按最优的方法（即每次取编号最大的）会 WA，原因是我们可能可以通过访问编号较小的来获得更优的编号更大的点，再访问这个点的话，就能取到更优的情况。因此我们可以用 set 维护，先判断能取的里最小的是不是不会产生贡献，如果不产生贡献就先把它取了，否则取编号最大的。

T3 solution

observation1: 最多只会有一个重合。

考虑一个合法选择，只需要找到 LIS 包含 a, b，LDS 包含 c, d，并且：

$$\max(a, c) < min(b, d), p_a < p_c, p_b > p_d$$

即可。

不妨设 $p_b > p_c$，那么考虑如果确定 c，d 一定是选择尽量靠右（因为这个时候 $p_b > p_d$ 一定成立，我们对 (c, p_c) 做扫描线，可以发现合法的 p_a 是一段左下角 。而这个时候 $a > c$ 一定成立，所以只需要对 a 找 $c < a < d$， p_c 最大的 c 即可，再找到 可能的最大的 p_b，比较一下就可以判断 a 上的结果。

$p_b < p_c$ 的情况，我们 reverse 一下即可。

这个东西是一个简单的扫描线。

T4 solution

链上的做法是对每个 L 维护点减边，因为点减边最少是一，所以实际上是统计的线段树上最小值个数。

树上做法需要改变维护的东西：

首先变成有根树，对于每个点它会 $+1$ 当且仅当 0 到 $L-1$ 中没有它的儿子。如果是一条从下向上的链那么直接查询为 1 的位置个数（维护最小值以及个数）即可。

现在来考虑路径的 LCA 。

首先我们可以轻松更新每个节点在 $0 \leq x < L$ 中的儿子个数是 2 的时间。

然后我们需要求出来一个长度的区间：满足恰好有一个在儿子个数 ≥ 2 的点。并且这个点的父亲还没有被选。将所有点按照儿子个数 ≥ 2 的最小长度加入堆，这玩意是很好找到的。

这段区间里面，不难证明没有儿子的点的个数至少是 2，并且当且仅当没有儿子的点的个数是 2 的时候权值为 0 到 $L-1$ 的点组成一条链。

复杂度 $O(n\log n)$。