吃糖果(eatcandy)

糖果盒其实就是一个队列。对于每种糖果，入队一次就要花费一点体力值，出队一次无代价。所以最少的体力值是n天出现的糖果的种类数。

如何求最少格子数？

维护一个队列，当第i天的糖果不在队列出现时(这个判断可以开个bool数组)，如果队首有糖果，并且这个糖果在未来n-i天都没再出现过，那么这个队首就可以删除，然后在队尾添加第i天的糖果种类，队列长度不变；否则，直接在队尾添加第i天的糖果种类，队列长度+1。最后队列的长度就是最少格子数。

如何求判断未来n-i天第i个糖果都没再出现过？

这个一开始预处理一下，开个数组记录每种糖果最后一个出现在哪一天。

求和(sum)

解法1:

O(n logn)递归模拟一遍求出数组b，暴力询问，期望得分20.

解法2:

O(n logn)递归模拟一遍求出数组b，用可持久化线段树做询问，期望得分40，但对大数据无效，不是一个好的思考方向。

解法3:

找到递归的规律，比如n=5，按层数写出来如下：

[1,2,3,4,5]

[1,3,5] [2,4]

[1,5] [3] [2] [4]

发现每一层的一段数组，其实是一个公差是2^i(i是层数，从0开始)的等差数列，所以对于询问的x, y，可以用像线段树的那样写法去找出最多2*logn个等差数列区间来计算。直接求和的话，可以拿到大数据的30分($u = 1, v = n$)。每个区间的计算，找到在$[u, v]$范围内的等差数列，就可以算出答案。

期望得分100。

本题来源于cf的117D，原题n的范围是1e18，有取mod操作。因为等差数列计算公式会用到除法，取mod会有不同，所以本题删除了取mod，并且数据范围减小防止暴long long。

寻找魔法师(wizard)

这题是一个隐藏的博弈问题。Idea来源于cf gym 101431E。

这题有个误区，就是需不需要知道所有邻居的大小关系，才能在所有的情况下保证一定找大一个极大值。答案是否的。

因为这是一个无环的结构，你中途调查的结果会影响你后面调查的决策。

假设T(i)表示第i个魔法师的魔法等级。

就以样例3来说，如果你第一次调查3号住宅，那么就有以下4种情况：

T(2)<T(3)>T(4) 这是最理想的情况，调查一次就找到了极大值，花费是4。

T(2)>T(3)>T(4) 遇到这种情况，接下来该往哪边调查？答案是调查1,2那边。因为调查4,5有可能找不到极大值，只要T(4)>T(5)；而无论T(1)<T(2)还是T(1)>T(2)，调查1,2这边一定能找到极大值，因为有T(2)>T(3)这条信息。因此，接下来只要调查1号住宅就可以找到极大值，花费是4+1=5。

T(2)>T(3)<T(4) 1,2和4,5都一定存在极大值，调查住宅1，花费是4+1=5

T(2)<T(3)<T(4) 1,2可能不存在极大值但4,5一定存在极大值，调查住宅4，花费是4+2=6。

以上4种情况，最坏的情况是第四种，因为要保证找到，所以答案是最坏的那一种。

这个题，可以看成，A想要花费最少代价找到极大值，B想让A找到极大值的代价最大。A每次可以询问i和i-1,i+1的大小关系，这个关系由B告诉他。两者都按照最优策略进行。

显然B不会直接告诉A T(i)比T(i-1)、T(i+1)都大，这样A就直接找到了极大值；

B也不会两者都小于T(i)，这样A无论在哪一边找，都可以找到极大值，那么A就会选择代价比较小的一边。

所以B必须选T(i-1)或T(i+1)之一令其大于T(i)，另外一个小于T(i)，为了让A花费更大的代价，B会选择代价比较大的一边，令其大于T(i)。

而A能做的是，选择调查的i是哪个，可以让最坏情况(B决定)最小。

为什么大于T(i)的那一边一定有极大值，小于T(i)那一边存在可能没有最大值？如果是一个环，这个结论还成立吗？这个留作思考，画一画例子。

解法1:

n=2输出min(a[1],a[2])；n=3输出min(a[1]+a[3],a[2])；期望得分20

解法2:

全部调查代价都是1的情况，每次找区间的中心调查，最终的答案是logn下取整。结合解法1期望得分是20+40=60

解法3:

使用区间dp，设状态dp[le][ri]是在第le到ri个住宅中调查找到极大值的最少花费(此时要求le,ri要么是边界，要么分别比le-1,ri+1的魔法等级高)

有如下状态转移方程：

$$\mathrm{dp}[le][ri] = min_{i=le}^{ri}(a[i] + \max(dp[le][i-1], dp[i+1][ri]))$$

边界条件：dp[le][ri]=0 (le>ri)

可以套区间dp模板实现，也可以记忆化搜索实现。复杂度是O(n^3)

期望得分100