A

dp

dp[i]表示处理完前 i 个点的最大收益。

有两种情况：

（1）第 i 个点不投资，那么 dp[i] = dp[i - 1]

（2）第 i 个点投资，那么 dp[i] = max(dp[k] + get(k + 1,i) - sigma(c[j]) (k + 1 <= j <= i))(0 < k < i)

这个的意思是，[k + 1,i]点选择全部投资，所以 sigma(c[j])表示每个点都投资了，get(k + 1,i)的意思是，[k + 1,i]完全包含了的给出区间的总价值

第二步需要优化，我们发现，任意一个 dp[i]的转移 dp[k]，sigma(c[j])，即 dp[k] 减去了从 k + 1,i 的每个数 c[i]，所以每做到一个数 i，给 1..i - 1 中的每个 dp 值减去 c[i]（因为以后要用某个 dp 转移的话，后面的每个数都被减了一次)。 第二个是收益[Li,Ri,Pi]，同样的方法，选择了 dp[k],k < Li 的话，意味着 get(k + 1,i)增加了 Pi，所以给 1...Li - 1 的每个 dp 值加上 Pi

思考：这题并查集是可以过的，FZU 某场月赛结束后 SCUT_DELL 的方法是并查集

B

基本都过了的题，部分和，然后二分正方形大小，枚举正方形的起点，判断是否有某个正方形的和<=limit

C.

方法一：树链剖分最入门的题目，有兴趣可以学一下，这里不做解释，复杂度 nlognlogn

方法二：随便选一个点作为 root，dfs 一次求出 root 到某个点的距离，并且得到 dfs 序。

一个显然的事实：dis(u,v) = dis(root,u) + dis(root,v) - 2 * dis(root,lca(u,v))，因此任意两个点的距离可以直接用到 root 的军理算出来。dis(root,i)表示从根到 i 的距离，dis(u,v)是 u 到 v 的距离

修改操作是，如果给(fa(u),u)这条边增加了一个值 x，那么 dis(root,u)增加 x，且其后代每个点的 dis(root,i)都增加了 x，这个操作，直接用得到的 dfs 序维护即可！!

D.

建树后，因为只有 100 个操作，直接模拟就行了。

E

第一场某道题的弱化版

10 次修改，直接每次修改暴力，故第二问可以忽略，重点是第一问。

用单调栈处理出每个数做为最小值，可以延伸到左边最左的位置，和右边最右的位置

然后可以得到每个值作为最小值的最长延续区间。直接将(值，长度)按照值排序，然后一个后缀 suf[i] = max(len(i),suf[i + 1]))表示不小于排序后的第 i 个数的长度最大值。每个查询，二分一下

这题可以并查集做，将查询和给出的序列的值从大到小排序，序列中每插入一个数，假设原位置 pos，看看集合 pos - 1,和 pos + 1 是否可以联合即可，查询即取当前时刻集合元素的最大值

F

挺难解释的一道题，随意讲一下。

状态压缩 DP，因为行数只有 3 行，所以按列来做。

Dp[i][j][k]表示做完前 i 列，第 i 列的状态为 j，且选了 k 个不相交矩形的最优解，j 的状态我在做的时候设了 13 种，表示第 i 列的状态，比如可能 3 行都不属于任意一个矩阵，有一行属于某个矩阵，或者两行。新加一列的时候，可以选择增加矩阵，或者延伸原来的矩形。

要写想得舒服，这题状态要设得好一点，否则状态会很麻烦。

G

基本都过了，直接暴力(dfs or 状态压缩)枚举每个?填什么值，然后判断括号序列是否合法即可。复杂度 2^16

数据范围放大到 1K 左右的时候，dp 做亦可。

注意到一个事实，对于任意一个合法的括号序列，左括号和右括号数目相等，在此前提下，然后一个右括号左边的左括号数目不小于当前右括号的数目，那么一定匹配

Dp[i][j]表示处理到字符串的第 i 位，左括号的数目，下一个(，变成 dp[i][j + 1],如果下一个是)，那么变成 dp[i][j - 1](j >= 1)。如果问号，枚举是什么就行了。最后求 dp[strlen(s)][0]即是结果