A：摸个规律，答案只能是1,1,1,/1,1,2 ,所以3的倍数/4的倍数就直接算一下，否则没有解。

B：按照一种不能被证明的排序，每个节点保存l,r,delta, 分别代表内部相消之后左边还能匹配多少右括号，右边还能匹配多少左括号，以及二者的差，按照差>l>r的优先级进行排序。

C：注意到不共线，排序后三个三个的输出。

D：可以用优先队列保存当前可以使用的数字，然后双指针扫区间就行。。

G：好像规律比标算nb啊，找到的规律是这个序列干掉第一个1，其他的就有x个2他就出现了x+1次。Loglog还不到400ms

H: 这个比赛没做出来（T了），卡了我nlog的优质算法（大雾）。然后这个就是建一种叫做笛卡尔树的treap的一种，用于在一个序列上平地拔起一根treap，然后操作太高级没学会，只会还原原序列（误），还会在树上dfs还是偷的杜教的板子（x）.

有了这个笛卡尔树之后，我们先搞好A的笛卡尔树，然后考虑B，B和A是所谓的rmq similar， 那么这两个序列基于权值和下标搞出的笛卡尔树应该是一样的，那么也就是数一下B数列有多少能搞到这个相同的笛卡尔树，笛卡尔树本身是按照value进行奇怪的排序，我们要求的实际上就是这个树的拓扑排序的个数。那么怎么求呢（好吧我不会），那么结论是 , ~~好了用这个数乘每个序列的期望就做完了~~  首先对于n=1,结论成立，那么对于非叶子节点x， 假设他左儿子sz = n, 右儿子sz = m, 那么实际上我们可以直接得到f (x) = f(ls(x) ) \* f(rs(x)) \* (n + m)! / (n! \* m!), ls代表左儿子，rs代表右儿子，按照归纳证明，将f(x) = ,带进f(ls(x)) 和f(rs(x))一顿乱消就能得到(n+m)!/ ∏ sz(ls(x))\* ∏sz(rs(x)) , 上下同乘(n+m+1), 就发现式子对f(x)成立，归纳完毕。然后有了这个式子我们瞎几把树dp就行了。