20231010题解

T1、SUM凑和

1~10：才5个数，暴力枚举所有情况即可。这就40分了啊。(205<108)

11~12：10个数但是都很小，这种情况下如果能改到的话最多改2次，暴力枚举不改、改1次、改2次的情况就行了。

13~15：数比较小，不确定剪枝的暴力能不能过。

16~17：其实这里就不好办了。参考正解。

18~20：这个特殊性质其实没啥用，只是简化了分组背包的一些情况。

100pts：

分组背包。

初始已经装了s=sum{a[i]}的背包，要给每个元素改变一下重量，使得总重量=m，且改动次数最少。

注意到对每个元素，改动的总情况数是19种，加上不改的情况，总共20选1，直接用分组背包。

设dp[j]表示恰好装满j的背包至少要改多少次，显然s>j时不可能做到，初始dp[s]=0。

直接对每组内更新dp[j]=dp[j-w[k][i]]+1即可，w[k][i]为第k个物品做第i种修改时改变的重量。

时间复杂度O(nm)，常数20，可以通过。

问题：本题是否有更好的解法？

T2、GCD矩阵

1~2：送分，直接找全2矩阵就可以了。

3~5：可以枚举所有矩形用GCD找最大公约数。O(n6)能过。

6~10：这里开始就要枚举公约数了（注意这里没有单调性不能二分）。将矩阵内所有元素对公约数取模，就变成了找最大面积的全0矩阵。就变成了P5943【最大的园地】。

这个数据范围可以用二维前缀和优化暴力来找全0矩阵，时间复杂度O(n4k)。

11~12：不用枚举公约数，直接模2二维前缀和优化就可以O(n4)过掉。

16~17：优化一下常数说不定能直接过？

18~20：变成了一维问题，直接暴力枚举区间就可以O(n2)。

100pts：

悬线法模板。

记up[i][j]为(i,j)出发往上走连续最多的0的个数。left[i][j],right[i][j]表示从(i,j)出发往左往右的0的最左最右坐标。

我们考虑一段竖直的连续0，它能构成的最大矩形的**宽度**为这些元素中，min{right}-max{left}+1。因此可以在遍历的时候直接统计。

递推：left[i][j]=max{left[i][j],left[i-1][j]}，right[i][j]=min{right[i][j],right[i-1][j]}。

当然只在(i,j)和(i-1,j)都是0的时候才统计，到(i,j)的最大矩形面积为up[i][j]\*(right[i][j]-left[i][j]+1)，直接比较即可。

时间复杂度O(nmk)。k=max{a[i][j]}。

是否有时间复杂度和k无关的做法？

或者你可以用单调栈扫一遍，因为每一列都是一些高矮不一的长条矩形。

T3、MIN费用

原题P6869

1：n是5，直接暴力都能过

2：一条链，直接O(n2)跳一跳也过了。

3：一条链还是顺次相接的，直接边权求和秒了。

4~7：n=10，暴力可能差一点(n^n有点大了)，但是树结构比较好的话暴力能卡过去

8~12：O(n2)能过，DFS写好一点就行了。

13~17：一条链，但是不能直接O(n2)跳，考虑存下每个点的深度，利用比较深度来确定走的方向，差分来维护每条边经过多少次，O(n)能过。

18~20：一个DFS序为1~n，则有1->2->3->...->n->1恰好经过每条树边2次，则我们找出1->n的路径，将每条边走2次的最优方案减去1->n的最优方案即为答案。O(n)。

100pts：

直接DFS肯定超时。考虑一些比较快的方法。

题目要求找到1->2->3->...->n的路径并计算费用，那么树上路径当然用倍增LCA来找。

每次找到路径之后让路径上的边的使用次数use[i]+=1，最后取min{c1[i]\*use[i],c2[i]}即可。

当然要是直接use[i]+=1也会超时，正确做法是树上差分，对i->i+1的路径，令t[i]++,t[i+1]++,t[lca(i,i+1)]-=2即可。之后做一次子树和就可以得到use数组了。

时间复杂度O(nlogn)。

当然树剖也可以做。

T4、LCM区间

原题P6659

1~2：z=1且M≤100，直接暴力枚举区间左右端点都可以做。O(M3logM)。

3~5：z=10且M≤1000，考虑枚举左端点和区间长度，当LCM>M时直接退出该端点的枚举，理论最坏复杂度O(M2zlogM)，能过。

6~7：一个数但是1e9，注意到假设区间长度为2，则如果存在[x,x+1]，则M=x(x+1)，直接判断[sqrt(M)]是否满足条件即可。也可以二分找。区间长度不小于3时最大数为1e3，这样就可以将时间复杂度压缩至O(M2/3logM)。

8~10：100个1e9，方法类似，事实上由于区间长度变大时右端点变小，因此复杂度到不了上面的最坏情况，卡卡常就过了。

11~12：一个1e18，直接预处理106以下情况就行了。

13~16：考虑分解质因数，如果存在这样的区间，那么最大质因子一定包含在内，如果次数是1次，那么可以将区间长度控制在该质因子以内且必须包含这个质因子，再结合其他的大质因子可以减少枚举量。

17~18：质数只有2输出1 2其他都输出NIE。

100pts：

数学题，有点难想。

假设区间长度为2，则如果存在[x,x+1]，则M=x(x+1)，直接判断[sqrt(M)]是否满足条件即可。

假设区间长度不小于3，则最大数不超过106。

又因为19!<1018<20!，故区间长度不超过19。因此对区间长度大于2的部分可以直接预处理，用map将M值对应到区间的左右端点即可。枚举左端点和区间长度，利用gcd把lcm搞出来对应一下就行。由于左端点从小到大枚举，所以预处理存第一个就行，最后和区间长度为2的方案比较一下就行了。

时间复杂度O(z)或O(zlogM)加上前面预处理的复杂度，3s内足够了。