T1取数博弈

20pts

if,else判一下

30pts

暴搜一下，或者胡乱贪心都可以。

40pts

因为所有数字都相同，所以最优操作显然是每次去掉中间那个和直接取一半。递归dfs计算一下即可。

100pts

区间DP，考虑两个人在一轮中的操作，先手（删除操作的人）想要让得分最小化，后手（取数操作的人）想要让得分最大化，所以显然有如下的转移方程：

$dp(l,r) = min_{i=l}^{r}\{max(sum(l,i-1)+dp(i+1,r), sum(i+1,r)+dp(l,i-1))\}$

边界条件为dp(i,i)=0，非法的dp与sum可直接置零。

T2货物收集

部分分设置

对于30%的数据，给阶乘级别的搜索算法。

对于60%的数据，考虑对所有边权排序，然后每次增加的量肯定是某两个边权的差值。模拟增加过程，当满足条件时输出即可。时间复杂度$\mathcal{O}(n^2)$

对于链上的情况，贪心向两端扩展即可。

题解思路

我们考虑经过一条边并不会减少我们的武力值。

我们只需要二分一个我们的武力值，然后O(n)判断一次当前能取到多少货物，就可以了。时间复杂度$\mathcal{O}(nlogv)$

T3 货物分组

Notice: 这道题在考试过程中，为了卡掉一些剪枝的暴力，不小心把错误的贪心全部放过去了，正在重造数据。
UPD: T2数据已经重新构造，并且已经重测。为大家带来的不好体验，我表示万分抱歉。

部分分设置

对于10%的数据，我们暴力枚举分组数以及分组情况。

对于30%的部分分，留给n^3暴力DP

对于60%的部分分，我们稍后会介绍一个n^2暴力DP

正解思路

首先考虑一个DP

$f[n][m]$表示前n个分成m组的最小花费。我们只要枚举k之后
$f[n][m]=minf[k][m-1]+m*(s(n)-s(k))+max(nk+1)-min(nk+1)$
其中$s(n)$表示物品代价的前缀和。
这样转移是n^3的。我们考虑使用先付代价来DP.

$f(n)$表示前n个分成若干组的最小代价。那么我们枚举k之后

$f(n)=min\{f[k]+cost(k+1,n)+sum\_all-sum(1\ n)\}$

其中$cost(l,r)$表示l~r的重量和加上其中最大值减最小值。

这样是n^2的。

我们发现我们在转移过程中，可以用一个单调栈来维护最大值和最小值的变换，然后用线段树维护区间最值，就能每次$\mathcal{O}(log\ n)$更新答案。

时间复杂度$\mathcal{O}(nlogn)$

T4 地形计算

部分分设置

对于30%的分数，给直接$\mathcal{O}(n^4)$暴力枚举点然后判断的做法。

对于60%的做法，给$\mathcal{O}(n^2)$ 或可能存在的 $\mathcal{O}(n^2poly(log(n)))$

解题思路

首先我们考虑一个很Naive的暴力，从每个点往下搜四层，如果回到了这个点，那么说明有一个四元环，我们把中间经过的点累加进答案。

这样每次扩展n 个点，扩展四次，时间复杂度$\mathcal{O}(n^4)$，如果你的实现比较美观的话，是$\mathcal{O}(n^3)$的，不过因为算法本质相同，并没有设置这两种实现方式的分数区分。

我们考虑另一种做法，不扩展四层，而是每次扩展两层，用一种MIM(Meet In Middle)的方式统计答案，时间复杂度就降到了 $\mathcal{O}(n^2)$，可以拿到60分。

似乎已经没有更好的方法了，但是我们考虑对于60分的做法，我们切换枚举顺序，从度数最大的点开始枚举。

每次枚举完毕以后，删去度数最大的点，然后重复这个过程，也能统计出答案。正确性显然，我们考虑这个做法的复杂度。每次都使nnn减小1，那么复杂度为$\mathcal{O}(n+(n-1)+(n-2)+...+1)=\mathcal{O}(n^2)$ O(n+(n−1)+(n−2)+...+1)=O(n2) 看起来没什么用，只是多了一个12\frac{1}{2}21的常数。

~~但是因为Venn在这种做法上施加了魔法，所以复杂度是正确的。~~

其实我们上述复杂度的分析，给出的复杂度上界达不到，我们考虑$O(n^2)$并不是他的时间复杂度。我们换一种分析方式，对于选择的每一个度数小于$\sqrt{m}$的点，他向外会扩展最多$sqrt(m)$次。对于度数大于$\sqrt{m}$的点，他会向外扩展多次，但是由于所有点的度数之和不能超过2倍的m,所以单次均摊是m次扩展，这样的点不会超过$\sqrt{m}$个，因为所有点的度数之和等于$2m$。那么总时间复杂度就为$\mathcal{O}(m\sqrt{m})$