

美丽数

- 考点：贪心、分类讨论
- 难度 CSP-J T2

测试点 1~2

给爆搜的部分分。

测试点 5~6

特殊部分分。

当 a 中只有两个数的时候，设分别是 x, y ，那么最终的美丽数只有可能是 $xyxy \dots xyx$ 、 $xyxy \dots xy$ 、 $yxyx \dots yx$ 、 $yxyx \dots yxy$ 其中之一。

分类讨论一下即可。

测试点 7~8

a 中只有三个数非 0，设分别是 x, y, z 。

贪心的考虑，我们从数位高到低来考虑，那么我们肯定会先放最小的（如果前一个位放了最小的那就放次小的）。

但什么时候不能这样放呢？就是剩下的 x, y, z 不能形成一个美丽数了。

考虑一下就能知道 x, y, z 能形成美丽数的充要条件就是 $a_x \leq a_y + a_z + 1$ ，我们设 $a_x > a_y > a_z$ 。

所以判断一下就可以了。

ps：如何处理不能有前导 0？可以设前一个放的数是 0。

测试点 3~4

可以沿用上一档部分分的做法，从数位高到低考虑，对每一位依次考虑 $0 \sim 9$ 如果放在这个位了之后，剩下的数还能不能形成一个美丽数。

由于 n 不大，所以我们可以暴力的去判断，复杂度为 $O(n^2)$ 。

测试点 9~10

关键在于如何判断当前的数还能不能形成一个美丽数。

我们设 $mx = \max\{a_0, a_1, \dots, a_9\}$, $su = \sum_{i=0}^9 a_i$, 那么能形成的充要条件就是 $mx \leq su - mx + 1$ 。

具体写法可以参考std。

ps: 你有没有考虑输出为 0 的情况? 虽然数据里是没有的 (不能太刁钻了), 但还是要注意的。

比赛

- 考点: 排序、树结构。
- 难度 CSP-J T2+

测试点1~4

留给暴力的做法。

测试点 5~10

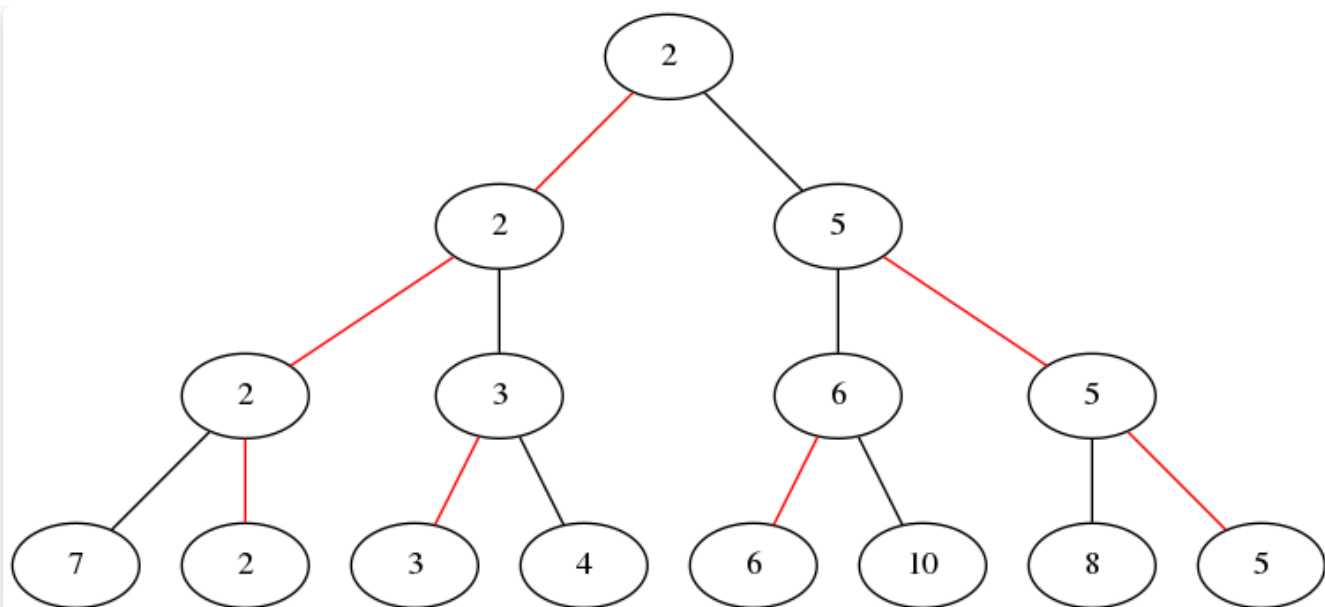
也是暴力做法, 其实就是模拟题意。

暴力模拟一次所有比赛过程的复杂度是 $O(n \log n)$, 所以总复杂度就是 $O(n^2 \log n)$ 。

测试点11~14

发现上面暴力做法的瓶颈在与我们不知道比赛的每个过程是什么样, 也就不知道选手 i 在他本应输的那一局赢了后会带来什么影响。

其实这个比赛过程是有名字的, 就是[锦标赛排序](#)。整个比赛过程看成是一棵二叉树, 通过拿下面的例子来分析:



其中最底下的节点就表示所有选手，上面的每一层就代表了一场比赛中的胜者。

所以你会发现，如果选手 i 在他本应输的那一局赢了后，对其他的比赛都没有影响。那么我们可以将整个树建出来后（也可以不显示的建树），从当前节点依次向父节点查看。

因为 a_i 是递增的，所以一旦当前节点是父节点的左儿子，那么就一定会小于那个右儿子的权值。

测试点15~20

现在考虑 a_i 不递增的情况。

那么我们可以在每个节点记录上子节点中权值较小值 a_i ，和较大值 b_i 。

那么设当前要处理的是选手 i ，我们先要找到选手 i 第一次会在哪输，那就是第一个 $a_x > a_i$ 的祖先节点 x 。

之后考虑第二次输在哪？就是 $b_y > a_i$ 的那个节点，其中 y 得是 x 的祖先节点。

所以就可以得到答案了，复杂度是 $O(n \log n)$ 。

他会输出啥

- 考点：简单模拟，字符串的输入，简单数学计算（等差数列）。
- 难度：CSP-J T3+

一、读入

我们发现其实 A 和 B 是什么我们并不关心，即使 d,e 会出现 A 对应的字母，但只要我们判断一下 d,e 是不是字母就可以了（而不用具体关心是什么字母）。

所以我们真正关心的只有这一部分：

```
1
2         a,b,c
3             d,e,f
4
5
```

std 使用的是 `getchar()` 来读入，那么就有一下代码：

```
1  int read1(){
2      int sum=0,f=1;char st=getchar();
3      if(st=='-')f=-1,st=getchar();//判断是否是负数
4      while('0'<=st&&st<='9'){
5          sum=sum*10+st-'0';
6          st=getchar();
7      }
8      return sum*f;
9  }
10 int read2(){
11     int sum=0,f=1;char st=getchar();
12     if('a'<=st&&st<='z'){
13         st=getchar();
14         return 1e9;//1e9代表是 A
15     }
16     if(st=='-')f=-1,st=getchar();
17     while('0'<=st&&st<='9'){
18         sum=sum*10+st-'0';
19         st=getchar();
20     }
21     return sum*f;
22 }
23 int main(){
24     int ch=getchar();
25     while(ch!='(')ch=getchar();
26     int a=read1();
27     int b=read1();
28     int c=read1();
29     ch=getchar();
30     while(ch!='(')ch=getchar();
31     int d=read2();
32     int e=read2();
33     int f=read2();
34 }
```

二、range(a,b,c) 的处理

- 若 $c > 0$ ，那么 `for A in range(a,b,c):` 就等价与 `for(int i=a;i<b;i+=c)`
- 若 $c < 0$ ，那么 `for A in range(a,b,c):` 就等价与 `for(int i=a;i>b;i+=c)`

若我们暴力枚举，复杂度就是 $O(|ae|)$ 。

但要注意，第二个循环实际上求的就是等差数列的和。

所以我们可以用 $ans = \frac{1}{2}(\text{首项} + \text{末项}) \times \text{项数}$ 来快速求解。

游览计划

- 考点：图论，bfs，贪心。
- 难度：CSP-J T4

测试点 1~3

给暴力留的分。

测试点4~8

先通过bfs求出每个点到另一个点的最短路径的长度 $d_{i,j}$ ，然后暴力枚举四个点 a, b, c, d ，然后求 $\max(d_{a,b} + d_{b,c} + d_{c,d})$ 就可以了。

复杂度 $O(n^4)$ 。

测试点9~10

答案就是 3，因为在完全图中，任意两点的最短路都是 1。

测试点11~15

同样求出 $d_{i,j}$ ，然后暴力枚举三个点 a, b, c ，然后求出离 c 点最远的点 f_c ，但是 f_c 可能是 a 或者 b ，所以需要求出离 c 点第二远的点 g_c ，第三远的点 h_c ，这样就一定可以找出那个最远的点。

复杂度是 $O(n^3)$ 。

当然也有许多其他做法。

测试点16~20

优化上一个算法。

同样求出 $d_{i,j}$ 和 f_i, g_i, h_i 。

我们枚举两个点 a, b ，然后经过 $a \rightarrow b$ 的最远路径一定是 $f_a/g_a/h_a \rightarrow a \rightarrow b \rightarrow f_b/g_b/h_b$ 其中之一。

依次考虑这几种情况，排除掉非法的情况（经过相同的点），直接枚举就可以了。

复杂度是 $O(n^2)$ 。