

# 2020牛客暑期多校训练营（第五场）

## 比赛情况

题号	A	B	C	D	E	F	G	H	I	J	K
状态	-	-	-	0	0	0	-	-	0	-	-

0 在比赛中通过 0 赛后通过! 尝试了但是失败了 - 没有尝试

## 比赛时间

2020-07-25 12:00-17:00

## 题解

### D - Drop Voicing

定义两种移动，分别是

Drop-2 把当前倒数第二个数字拿到最前面来。

Invert 把当前最前面的数字放到最后去

我们定义若干个Drop-2移动是一次操作，问最少要几次操作才能使得序列有序。

猜了个结论，把序列二倍之后以每一个位置为递签，求长度为n的序列中的最长不下降子序列，保留最大答案，最后的答案就是n减去这个答案。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N = 1005;
int f[N], n, a[N];
int main()
{
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
    {
        scanf("%d", &a[i]);
        a[i+n] = a[i];
    }
    int tans = n;
    for (int i = 1; i <= n; i++)
    {
        int ans = 0;
        for (int j = 1; j <= n; j++) f[j] = 1;
        for (int j = i; j <= i+n-1; j++)
        {
            if (a[j] > a[j+1])
                f[j+1] = 0;
            else
                f[j+1] = f[j];
        }
        for (int j = 1; j <= n; j++)
            if (f[j] == 1)
                ans++;
        tans = min(tans, ans);
    }
    cout << tans;
}
```

```
{  
    for (int k = i;k<j;k++) {  
        if (a[j] > a[k])  
            f[j-i+1] = max(f[j-i+1],f[k-i+1]+1);  
  
    }  
    ans = max(ans,f[j-i+1]);  
}  
tans = min(tans,n-ans);  
}  
printf("%d\n",tans);  
return 0;  
}
```

## E - Bogo Sort

给一种置换 \$P\$ 问有多少种排列可以通过多次这个置换变成单位置换。

相当于求置换 \$P^k\$ 有多少种不同的值，显然答案就是 \$P\$ 的每个循环节的 \$\text{lcm}\$ 然后用 \$\text{py}\$ 水一下高精度就可以了。

```
def gcd(a,b):  
    if(b==0):  
        return a  
    else:  
        return gcd(b,a%b)  
  
n = int(input())  
p = [0] + [int(x) for x in input().split()]  
vis = [0 for i in range(n+1)]  
  
ans = 1  
  
for i in range(1,n+1):  
    if(vis[i] == 0):  
        u = p[i]  
        l = 1  
        vis[i] = 1  
        while(vis[u] == 0):  
            l+=1  
            vis[u] = 1  
            u = p[u]  
        ans = ans*l//gcd(ans,l)  
  
ans = str(ans)  
if len(ans) > n:
```

```
ans = ans[-n:-1]
print(ans)
```

## F - DPS

签到题，记录最大值，输出一个50乘以当前值除以最大值长度的方块，最大值对应的方块要在里面画个星星，模拟即可。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N = 105;
int d[N];
int maxd = 0;
void output(int x,int id) {
    printf("+");
    for (int i = 1;i<= x;i++)
        printf("-");
    printf("+\n");
    printf("|");
    for (int i = 1;i<= x;i++)
    {
        if (i==x && d[id]==maxd) printf("*");
        else printf(" ");
    }
    printf("|%d\n",d[id]);
    printf("+");
    for (int i = 1;i<= x;i++)
        printf("-");
    printf("+\n");
}
int main()
{
    int n;
    scanf("%d",&n);
    for (int i = 1;i<= n;i++)
    {
        scanf("%d",&d[i]);
        maxd = max(maxd,d[i]);
    }
    for (int i = 1;i<= n;i++) {
        double tmp = (double)d[i]*50;
        tmp = tmp/maxd;
        tmp = ceil(tmp);
        output(tmp,i);
    }
    return 0;
}
```



# 比赛总结与反思

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team



Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai\\_milk:20200725%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95&rev=1596122171](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:20200725%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95&rev=1596122171)

Last update: 2020/07/30 23:16