

20221115 测试总结

依旧是骗分ing

T1 Reverse

依旧是字符串。

刚开始题意理解错了，发现样例退不出来，才退回去读题，才把题意读懂了。原本是想打dfs的，结果炸了（下午改的时候发现应该是k打成n了，考试的时候调半天没有调出来，浪费了很多时间，还没有拿到分，下午改题的时候刚开始还是写错了，眼睛大概是瞎了吧，希望考试的时候不要这么瞎👁👁）。只好乖乖去打部分分，写了个特判，骗了15分。

后面改题的时候，发现我的思路其实和题解差不多，但是实现的方式不一样。把之前的搜索改了之后（还没有用set，会TLE），发现不仅TLE，还WA了，问了人之后才发现我实现的方法有问题，错在取边界的时候考虑不周到，画了图手推几遍才懂。这道题改了很久。（试图卡常但是失败）

```
//reverse
#include<cstdio>
#include<cstring>
#include<cctype>
#include<iostream>
#include<bitset>
using namespace std;
const int MAXN = 100100;
const int inf = 0x3f3f3f3f;

template<typename T>inline void read(T &x)
{
    int c=getchar();
    x=0;
    T neg=0;
    while(!isdigit(c)) neg|=(c^'-'),c=getchar();
    while(isdigit(c)) x=(x<<1)+(x<<3)+(c^48),c=getchar();
    if(neg) x=(~x)+1;
}

int n,m,k,s,ans[MAXN],cnt,l,r;
bitset<MAXN> vis;
int q[MAXN];

inline int abb(int x)
{
    return x>0 ? x : (~x)+1;
}

inline int maxa(int x,int y)
{
    return x>y ? x : y;
}

inline int mina(int x,int y)
{
    return x>y ? y : x;
}
```

```

}

inline void bfs()
{
    int now,last,ll,rr;
    register int i;
    q[++r]=s;
    while(l<r)
    {
        now=q[++l];
        ll=maxa(now-k+1,k-now+1);
        rr=mina(n*2-k-now+1,now+k-1);
        register int i;
        if(k&1)
        {
            for(i=ll;i<=rr;++i)
                if(!(abb(now-i)&1))
                    if(!vis[i])
                    {
                        vis[i]=1,q[++r]=i,
                        ans[i]=ans[now]+1;
                    }
        }
        else
        {
            for(i=ll;i<=rr;++i)
                if(abb(i-now)&1)
                    if(!vis[i])
                    {
                        vis[i]=1,q[++r]=i,
                        ans[i]=ans[now]+1;
                    }
        }
    }
    return ;
}

int main()
{
    freopen("reverse.in","r",stdin);
    freopen("reverse.out","w",stdout);
    read(n),read(k),read(m),read(s);
    register int i,j;
    int quq;
    for(i=1;i<=m;++i)
        read(quq),vis[quq]=1;
    if(k==1)
    {
        for(i=1;i<=n;++i)
            i==s ? printf("0 ") : printf("-1 ");
        return 0;
    }
    vis[s]=1;
    bfs();
    for(i=1;i<=n;++i)
        if(ans[i]==0) ans[i]=-1;
}

```

```
ans[s]=0;
for(i=1;i<=n;++i)
    printf("%d ",ans[i]);
return 0;
}
```

T2 Silhouette

这道题长得好像初中有一种题型，自己能手推样例，但是写不出来。于是去看下面的部分分，有 $n=1$ 的点，骗了5分。后面也有 $n=2$ ， $n=3$ 的点，但是想了很久还是没有写出来，总是感觉排除的时候很复杂，下午看的时候发现就是容斥定理没有学好（数学板块太差了），只好放弃去写下一题。

其实这道题是道数学题，可以将这道题转化为解方程。

T3 Silhouette

概率题，下意识地回避，先是输出了样例，再是骗了 $n=1$ 的分，就跳过了。但是其实题没有怎么读。

T4 Ancient

数学题。去除这道题复杂的题目背景，其实将很多数学知识糅合在一起，就是一个数学全家桶（但是我吃不下）。做题的时候只是知道考了一个组合数（可能是卢卡斯定理，但就是卡住了，写了个乱搞的组合数），还有一个逆元（但是用的是欧拉定理）。因为样例很小，当时过了，结果交上去就全WA了。

后面看题解的时候，一长串的数学知识，我震惊了(ΩДΩ)。只能去自学书上的数学板块（几乎就没有完全掌握的，数学实在是太差了QAQ）。现在还没有改出来。

总结：

这次考试还是只能去骗分，而且有些分还没有骗到。有很大板块是有关于数学的，就做得非常不好，只能是考试之前多多看书，但愿有用。还有就是第一题花费了很多时间，本来是一个小地方打错了，但是调试半天还没有调出来（虽然当时的思路也是有问题），时间分配应该要注意一下。这几天的改题，我也发现了其实听懂了时候自己打代码还是有些不够流畅，但是自己手推一遍之后再去打就会好很多，而且小细节出错的几率也不大，而且最近感觉打字速度要比CSP-S之前快了一些。

