



链滴

Password

作者: [xxjAc](#)

原文链接: <https://ld246.com/article/1545551222645>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



传送门

可将题目从后往前做

则问题变为将01序列转化为全是0的序列

由于对于当前序列不好操作

将当前序列差分

0表示与前一位相同，1表示与前一位不同

差分序列至多有20个1

操作变为反转两个之间距离为 $a[i]$ 的数

易证选两个0反转的操作是无意义的

而选一个1和一个0反转相当于将1左移或右移 $a[i]$

当可以反转两个1时即消灭了两个1

最终任务变为求消灭所有1的最小操作次数

可以bfs预处理消灭每两个1的最小代价

然后状压dp，msk表示没被消灭的1的序号

dp[msk]表示消灭这msk的最小代价

可枚举任意两个1消除来转移

程序：

```
//by xxj
#include<bits/stdc++.h>
using namespace std;
#define mp make_pair
#define ll long long
#define pii pair<int,int>
```

```

#define lowbit(x) x&-x
using namespace std;
const int inf=1e9+7;
const double eps=1e-10;
const ll linf=1e18+7;
const ll hh=523;
//const int mod=;
int n,k,l;
int x[17],a[107],cf[27],cnt=0;
int visited[10007];
int xx[10007];
int q[10007];
int wxt[27][27];
int dp[(1<<20)];
int go(int msk){
    if (dp[msk]!=-1){
        return dp[msk];
    }
    if (msk==0){
        return 0;
    }
    vector<int> v;
    for (int i=1;i<=cnt;i++){
        if (msk & (1<<(i-1))){
            v.push_back(i-1);
        }
    }
    dp[msk]=inf;
    for (int i=1;i<v.size();i++){
        if (wxt[v[0]+1][v[i]+1]==-1){
            continue;
        }
        dp[msk]=min(dp[msk],go(msk-(1<<v[0])-(1<<v[i]))+wxt[v[0]+1][v[i]+1]);
    }
    return dp[msk];
}
int main(){
    //freopen(".in","r",stdin);
    //freopen(".out","w",stdout);
    scanf("%d%d%d",&n,&k,&l);
    for (int i=0;i<k;i++){
        scanf("%d",x+i);
        xx[x[i]-1]=(xx[x[i]-1]+1)%2;
        xx[x[i]]=(xx[x[i]]+1)%2;
    }
    for (int i=0;i<=n;i++){
        if (xx[i]==1){
            cnt++;
            cf[cnt]=i;
            xx[i]=cnt;
        }
    }
    for (int i=0;i<l;i++){
        scanf("%d",a+i);
    }
}

```

```

}
for (int i=1;i<=cnt;i++){
    for (int j=1;j<=cnt;j++){
        wxt[i][j]=-1;
    }
}
for (int i=1;i<=cnt;i++){
    memset(visited,0,sizeof(visited));
    visited[cf[i]]=1;
    wxt[i][i]=0;
    int op=0,ed=0;
    q[0]=cf[i];
    while (op<=ed){
        int t=q[op];
        op++;
        for (int j=0;j<l;j++){
            if (t+a[j]<=n && !visited[t+a[j]]){
                if (xx[t+a[j]]){
                    wxt[i][xx[t+a[j]]]=visited[t];
                }
                visited[t+a[j]]=visited[t]+1;
                ed++;
                q[ed]=(t+a[j]);
            }
            if (t-a[j]>=0 && !visited[t-a[j]]){
                if (xx[t-a[j]]){
                    wxt[i][xx[t-a[j]]]=visited[t];
                }
                visited[t-a[j]]=visited[t]+1;
                ed++;
                q[ed]=(t-a[j]);
            }
        }
    }
}
memset(dp,-1,sizeof(dp));
int ans=go((1<<cnt)-1);
if (ans>=inf){
    puts("-1");
    return 0;
}
printf("%d\n",ans);
return 0;
}
/*
input:
*/

```