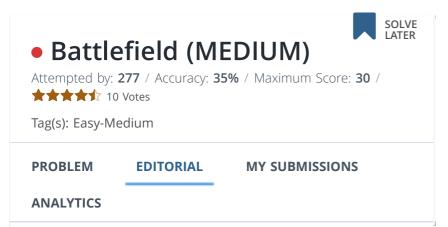
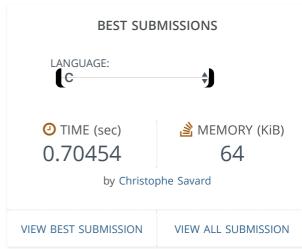
CHALLENGES PRACTICE COMPANIES

All Tracks > Data Structures > Arrays > 1-D > Problem



Our main task for this problem is to bring all K's and D's together in minimum number of swaps. For this we can count number of D's or number of K's in given input and then can construct window of that given length ,and for then all window can check how many D's or K's are missing to get the minimum number of swaps.





## **CONTRIBUTOR**



## **SOCIAL SHARE**

f y in G+

```
#define fe first
#define SZ 666666
#define si(n) scanf("%d",&n);
#define sl(n) scanf("%ld",&n);
#define pi(n) printf("%d\n",n);
#define pl(n) printf("%ld\n",n);
#define pf(n) printf("%f\n",n);
#define FILL(a,b) memset(a,0,sizeof(b));
#define rep(i,n) for(int i=0;i<n;i++)</pre>
#define reps(i,a,b) for(int i=1;i<=b;i++)</pre>
const int INF=1e9+5;
const int MOD=1000000007;
//_____***********
int main(){
  int T;
  scanf("%d",&T);
    while(T--){
         int N,k=0,countK=0;
         string slide;
         scanf("%d",&N);
         cin>>slide;
         slide+=slide;
         for(int
i=0;i<N;i++)if(slide[i]=='K')++k;
         for(int i=0;i<k;i++){</pre>
             if(slide[i]=='K')++countK;
         int maxK = countK;
         for(int i=k;i<(N+N);i++){</pre>
             countK = countK +
((slide[i]=='K')?1:0) -
((slide[i-k]=='K')?1:0);
             //printf("%d\n",countK);
             maxK = max(maxK,countK);
         }
       printf("%d\n",(k-maxK));
    }
return 0;
```