

题解

以下代码为标准输入输出！

t1

对于每一条狗，记录其上一次出现在马路哪边，判断一下即可。

代码：

```
#include<iostream>
#include<cstdio>
#include<cstring>

using namespace std;

int n,x,y,a[105],ans;

int main(){
    scanf("%d",&n);
    memset(a,-1,sizeof a);
    for (int i=1;i<=n;i++){
        scanf("%d%d",&x,&y);
        if (a[x]!=-1 && a[x]!=y) ans++;
        a[x]=y;
    }

    printf("%d\n",ans);

    return 0;
}
```

t2

解法一：枚举起点，向后累加，时间复杂度 $O(n^2)$ ，期望得分80。

解法二：从头开始累加，如果累加结果为负，则清零累加器，以下一个点作为开头累加，时间复杂度 $o(n)$ ，期望得分100。

解法二代码：

```
#include<iostream>
#include<cstdio>
```

```

using namespace std;

#define INF 0x3f3f3f3f

int n,x,tmp,ans;

int main(){
    scanf("%d",&n);
    ans=-INF; tmp=0;
    for (int i=1;i<=n;i++){
        scanf("%d",&x);
        tmp+=x;
        if (tmp>ans) ans=tmp;
        if (tmp<0) tmp=0;;
    }

    printf("%d\n",ans);

    return 0;
}

```

t3

解法一：暴力模拟，时间复杂度 $O(nm)$ ，期望得分30。

解法二：强行使用线段树维护每个小伙伴的糖，时间复杂度 $O(n\log_2 m)$ ，期望得分60（可能会被卡常）。

解法三：差分，化区间修改为单点，设 $f_i = a_i - a_{i-1}$ ，其中 a_i 表示第 i 个小伙伴的糖数，那么如果使区间 $[l, r]$ 增加 x ，只需将 f_l 增加 x ，将 f_{r+1} 减少 x 。最后 $a_p = \sum_{i=1}^p f_i$ 。时间复杂度 $O(2n + m)$ ，期望得分100。

解法三代码：

```

#include<iostream>
#include<cstdio>
#include<algorithm>

using namespace std;

#define LL long long
#define dd 1<<30

LL n,m,l,r,x,tmp,f[2000000];

inline int getint(){
    char ch,l; LL x=0;
    for (ch=getchar();ch<'0' || ch>'9';l=ch,ch=getchar()) if (ch==EOF) return dd;
    for (x=0;ch>='0' && ch<='9';ch=getchar()){if (ch==EOF) return dd; x=x*10+ch-'0';}
    return l=='-'?-x:x;
}

```

```

inline void putint(LL x){
    if (x<0) putchar('-'),x=-x;
    if (x<10){putchar(x+'0'); return;}
    putint(x/10); putchar(x%10+'0');
}

int main(){
    freopen("candy10.in","r",stdin);
    freopen("cnady10.out","w",stdout);

    n=getint(); m=getint();
    for (int i=1;i<=n;i++){
        l=getint(); r=getint(); x=getint();
        f[l]+=x; f[r+1]-=x;
    }

    tmp=0;
    for (int i=1;i<=m;i++){
        tmp+=f[i];
        putint(tmp); puts("");
    }

    return 0;
}

```

t4

此题暴力解法我可能不会，只提供正解。

由于休息时间，移动方向不只是向下或向右。

可以每个点拆点，将一个点拆为休息后移动1,2,3次到达这三个点，然后跑最短路即可。

点数为 $3n^2$ ，时间复杂度与最短路时间复杂度有关。

代码（sfpa）：

```

#include<iostream>
#include<cstdio>
#include<algorithm>

using namespace std;

#define LL long long
#define INF (LL)1000000000000

LL n,m,ans,t,w,x,y,nx,ny,step,tt,q[800030][4],f[300][300][3],a[300][300];
const LL dic[4][2]={{-1,0},{0,-1},{1,0},{0,1}};
bool vis[300][300][3];

```

```

void bfs(){
    t=0; w=1;
    q[1][1]=1; q[1][2]=1; q[1][3]=0; f[1][1][0]=0; vis[1][1][0]=1;

    while (t<w){
        t++;
        x=q[t][1]; y=q[t][2]; step=q[t][3]; tt=f[x][y][step];
        vis[x][y][step]=0;
        for (int i=0;i<=3;i++){
            int nx=x+dic[i][0],ny=y+dic[i][1],nstep=(step+1)%3;
            if (nx<1 || nx>n || ny<1 || ny>n) continue;
            if (f[nx][ny][nstep]>tt+m+(nstep==0?a[nx][ny]:0)){
                f[nx][ny][nstep]=tt+m+(nstep==0?a[nx][ny]:0);
                if (!vis[nx][ny][nstep]){
                    vis[nx][ny][nstep]=1;
                    w++;
                    q[w][1]=nx; q[w][2]=ny; q[w][3]=nstep;
                }
            }
        }
    }
}

int main(){
    scanf("%lld%lld",&n,&m);
    for (int i=1;i<=n;i++)
        for (int j=1;j<=n;j++)
            scanf("%lld",&a[i][j]);

    for (int i=1;i<=n;i++)
        for (int j=1;j<=n;j++) f[i][j][0]=f[i][j][1]=f[i][j][2]=INF;

    bfs();
    ans=min(min(f[n][n][0],f[n][n][1]),f[n][n][2]);

    printf("%lld\n",ans);

    return 0;
}

```