

[报告] C、Minimal Backgammon

[source]

<http://202.114.18.202:8080/judge/contest/view.action?cid=6151#problem/C>

[Description]

在一个从 0 到 N 的直线上走，起始点为 0，终止点为 N，每次掷骰子，然后走 1 到 6 格，如果当前点加上骰子点数超过 N，假如当前点为 i，骰子数为 j, $i+j > N$, 那么就走到 $2*N-i-j$ 点。此外在直线上存在两种点，一种 ‘B’ 点，走到它就跳回起点，然后继续当前回合。一种是 ‘L’ 点，走到它就是放弃当前回合，只能下一回合继续走。最后问在指定回合 T 内走到 N 的概率。

[方法]

这个可以很明显的用 $dp[i][j]$ 表示第 i 回合，到第 j 点的概率。然后面对当前点的三种不同情况进行 dp。最后求出 $\sum(dp[n][i])(1 \leq i \leq T)$ 的值即可

[Code]

```
#include<iostream>

#include<cstring>

#include<cstdio>

#include<map>

#include<algorithm>
```

```

using namespace std;

#define N 110

#define oo 100000000

double dp[N][N];

int vis[N];

int main(void)
{
    int n,t,l,b;

    while(scanf("%d%d%d%d",&n,&t,&l,&b)&&(n||t||l||b)){

        memset(vis,0,sizeof(vis));

        int x;

        for(int i=1;i<=l;i++){

            scanf("%d",&x);

            vis[x]=1;

        }

        for(int i=1;i<=b;i++){

            scanf("%d",&x);

            vis[x]=2;

        }

        memset(dp,0,sizeof(dp));

        dp[0][0]=1.0;

        for(int i=1;i<=t;i++){

```

```

for(int j=n-1;j>=0;j--){

    if(vis[j]==1){

        if(i>=2){

            for(int k=1;k<=6;k++){

                if(j+k<=n){

                    dp[i+1][j+k]+=dp[i-1][j]*1.0/6.0;

                }else{

                    dp[i+1][2*n-j-k]+=dp[i-1][j]*1.0/6.0;

                }

            }

            continue;

        }

    }

    if(vis[j]==2){

        dp[i-1][0]+=dp[i-1][j];

        continue;

    }

    for(int k=1;k<=6;k++){

        if(j+k<=n){

            dp[i][j+k]+=dp[i-1][j]*1.0/6.0;

        }else{

            dp[i][2*n-j-k]+=dp[i-1][j]*1.0/6.0;

        }

    }

}

```

```
        }
    }
}

double ans=0.0;

for(int i=1;i<=t;i++) ans+=dp[i][n];

printf("%.6f\n",ans);

}

return 0;

}
```