

Problem 1.1

- A winning strategy when the initial state of the switch is *off*.

Step 1: At beginning, the prisoners agreed to select one leader and made following decisions. The leader turned the light from "ON" to "OFF" whenever he found the light being "ON", while others turned the light from "OFF" to "ON" if the light was OFF and he/she didn't turn the light "ON" before.

Step 2: when the leader succeed turning the light "OFF" 'p -1' times, he declared that "*we have all visited the switch room at least once*" and stop.

Why this strategy works?

Proof: The light is turned on "p-1" times by the leader, which means the light has "p-1" times at "OFF" states, each "OFF" state can only be changed by one prisoner (not the leader). Each non-leader had only one chance to turn the light "ON". Thus all of the prisoner must be in the switch room once and turned the light ON.

Code explanations:

Key codes of leader thread:

```
int count = 1; // count the times of actions.
for(;count < NUMTHRDS;)//NUMTHRDS is the total number of prisoners
{
    pthread_mutex_lock (&mutexlight); //only one prisoner can touch the light
    if(light == ON)
    {
        light = OFF;
        count ++;
    }
    pthread_mutex_unlock(&mutexlight);
}
```

Key codes of non-leader thread

```
int flag =1, free=0; // each prisoner has only one chance to turn light
while( free ==0 ) // run forever if the leader didn't call "stop"
{
    pthread_mutex_lock (&mutexlight);
    if(light == OFF && flag > 0)
    {
        flag--;
        light = ON;
    }
    pthread_mutex_unlock(&mutexlight);

    pthread_mutex_lock (&mutexstop);
    if(stop == 1)
        free = 1;
    pthread_mutex_unlock(&mutexstop);
}
```

Experiment results

```
ji453898@cass:~/pthread/Question1.1
[ji453898@cass Question1.1]$ ls
Makefile  prisoner.c
[ji453898@cass Question1.1]$ make
g++ -pthread -o prisoner prisoner.c -I.
[ji453898@cass Question1.1]$ ls
Makefile  prisoner  prisoner.c
[ji453898@cass Question1.1]$ ./prisoner
time: 2.890000s
[ji453898@cass Question1.1]$ ./prisoner
time: 0.030000s
[ji453898@cass Question1.1]$ ./prisoner
time: 2.590000s
[ji453898@cass Question1.1]$ ./prisoner
time: 2.640000s
[ji453898@cass Question1.1]$ ./prisoner
time: 3.800000s
[ji453898@cass Question1.1]$ ./prisoner
time: 4.780000s
[ji453898@cass Question1.1]$ ./prisoner
time: 0.010000s
[ji453898@cass Question1.1]$ ./prisoner
time: 2.830000s
[ji453898@cass Question1.1]$ ./prisoner
time: 0.010000s
[ji453898@cass Question1.1]$ ./prisoner
time: 0.490000s
```