## The Underemployed Watchman

A certain watchman has only one duty, namely to shut down a reactor in case of a nuclear attack against the United States. He has never had to do his duty, and we all hope that he never will. Yet, he must stay alert!

The President has devised a system for keeping the watchman awake. He sits in a room with two buttons, one on the North wall, and one on the South wall. There is a light next to each button. When any light flashes, the watchman must press the button next to that light within 30 seconds.

There are also two chairs in the room, one near the North wall, and the other near the South wall. Neither chair can be moved. Since the room is very small, he can reach either button from either chair, but it is much easier to reach the button nearer a chair than the button further away. The watchman must be sitting in a chair at all times, except for the brief time it takes to change chairs. If a light flashes, the watchman can do one of two things: either press the button from the chair that he's in, or move to the other chair and then press the button from there.

The watchman wants to minimize "effort," defined as follows.

1. It takes no effort to press the button next to the chair where he's sitting.
2. It takes one "unit" of effort to press the button farther from the chair where he's sitting.
3. It takes one "unit" of effort to change chairs.

## Deterministic Algorithms

We proved in class that no algorithm for the watchman is better than 2-competitive. Here are three algorithms:

### The Lazy Watchman

The *lazy watchman* always stays in the same chair.

Prove that the lazy watchman is not competitive.

### The Eager Watchman

The *eager watchman* always moves to the chair next to the light that is on, if he's not already in it. He then presses the button from that chair.

Prove that the eager watchman is 2-competitive.

### The Cautious Watchman

The *cautious watchman* changes chairs if and only if the light on the farther wall flashes, and if the same light flashed in the previous step.

Prove that the cautious watchman is 2-competitive.

### The Optimal Deterministic Watchman

Prove that no deterministic watchman can be less than 2-competitive.

## Randomized Algorithms

### The Oblivious Forgetful Adversary

The *oblivious forgetful adversary* has no idea what the watchman is doing, and no memory of past requests. At each step, this adversary flips a coin, and chooses each request with probability one half.

Prove that every randomized watchman algorithm is $\frac{3}{2}$-competitive against the oblivious forgetful adversary.

### The Barely Random Watchman

The *barely random watchman* flips a coin, just once in his lifetime. With probability one half, he behaves as the eager watchman for the rest of his life; otherwise, he behaves as the cautious watchman all the rest of his life.

Prove that the barely random watchman is $\frac{3}{2}$-competitive against any oblivious adversary.

### The Forgetful Random Watchman

The *forgetful random watchman* stays in his chair if the nearer light flashes. If the farther light flashes, the forgetful random watchman flips a coin. With probability one half, he changes chairs.

What is the worst-case competitiveness of the forgetful random watchman against an oblivious adversary? (We always assume that an adversary is intelligent, in fact, has unlimited computational power, unless it is otherwise stated.)

### The Optimal Randomized Watchman

Find a randomized watchman strategy that achieves the lowest possible competitiveness against an oblivious adversary.