

## Instruction for Lab Assignment 2

### Problem description: Travel problem

There is a  $n$ -level building (from level 1 to level  $n$ ). Walking stairs from level  $s$  to level  $s + 1$  takes 1 minute. Taking a magic elevator from level  $s$  to level  $2s$  takes 1.5 minutes.

Example:

- 1). Walking from level 2 to level 3 takes 1 minute.
- 2). Taking magic elevator from level 2 to level 4 takes 1.5 minutes.

### Requirement:

Write python code to implement uniformCostSearch algorithm to solve this problem. Given a specified value  $n$ , output minimum cost and sequence of (action, newState, cost) from starting state 1 to end state  $n$ . For example, when  $n = 10$ , output should be formatted as follows:

Total cost: 5.0

('walk', 2, 1)

('elevator', 4, 1.5)

('walk', 5, 1)

('elevator', 10, 1.5)

### Hint:

The problem class and PriorityQueue class that will be used to implement uniformCostSearch have been provided in elevator.py and util.py respectively, where both files are allowed to be edited. The program should be developed based on these provided classes.