Hanyang Xiao
CS 350

Problem 2: Maze Solving

a) Input: A rectangular grid with height h and width w.
Output: the length of shortest path from (0,0) to (w-1,h-1)

```
count ← 0
while the front index in the queue is not (w-1,h-1), do
    deque the front index (i,j)
    if grid[i+1,j] is passable and unvisited
        add it into queue.
    if grid[i-1,j] is passable and unvisited
        add it into queue.
    if grid(i,j+1) is passable and unvisited
        add it into queue
    if grid(i,j-1) is passable and unvisited
        add it into queue
    count ← count + 1
if count < w+h
    return -1
else
    return count.
```

b) The input size is measured w*h where w is the width and h is the height.

Hence the worst case time complexity is O(w*h)

c) To find the shortest path between (0,0) ~~to~~ and (w-1,h-1), we need to create a stack

It doesn't affect the time complexity
But the space complexity will increase.

Hanyang Xiao
CS 350

Problem 1: Power Algorithm

a) Input: A non-negative integer n
Output: 2^n

If $n = 0$
    return 1
else
    return $P(n-1) + P(n-1)$

b) $A(n) = 2A(n-1) + 1 \quad A(0) = 0$

$= 2[2A(n-2) + 1] + 1 \quad = 2^2 A(n-2) + 2 + 1$

$= 2^3 A(n-3) + 2^2 + 2 + 1$

$= \cdots$

$= 2^i A(n-i) + 2^{i-1} + 2^{i-2} + \cdots + 1$

Let $n - i = 0$.

$A(n) = 2^n A(0) + 2^{n-1} + 2^{n-2} + \cdots + 1$

$= 2^n - 1$

c) It is not a good algorithm for solving this problem. A better approach could be that simply multiples 2 for n times.