CPSC 320 2024W1: Dynamic Programming Tutorial Problems

1 Legend of Zelda

[Thanks to Denis Lalaj for contributing this problem, which is also on assignment 4.]

Your friend is a big fan of the Legend of Zelda games and so decides to make his own version of them. He has been struggling to understand dynamic programming however, so he asks you to help him out with the knowledge you have learned from your CPSC 320 class. He seeks to set up the game in a 2D maze where Link starts his quest by entering the room with coordinates $(1, 1)$ in the maze, with the goal of saving princess Zelda who is trapped at the (m, n) corner. From each room, Link move to the neighbour room immediately to the right or down, but there is a caveat. Some rooms hide memories of Zelda (denoted by non-negative integers) that Link can collect, and these increase his health points. Other rooms hide monsters (denoted by negative integers), which damage Link's health points. To save princess Zelda, Link must ensure his health points do not drop to zero (or worse... negative) at any point. Your friend wants to add a super hard mode for the game, in which Link starts with the minimum possible health points that barely allows him to rescue Zelda, even if he chooses the best possible path.

Formally an instance of the problem is $G[1..m][1..n]$—a 2D array of size $m \times n$ where each entry is an integer value (representing the points in a room). For $1 \leq i \leq m$ and $1 \leq j \leq m$, let $HP[i, j]$ denote the minimum number of health points that Link needs to have, when starting by entering the room with coordinates (i, j). The problem is to compute $HP[1, 1]$.

Example: Two example grids are as follows, the first is 2×2 and the second is 3×2:

1	3
2	-4

-1	-1
3	2
1	4

For the grid on the left,

- $HP[2, 2] = 5$, since Link must have one point left after losing 4 points when entering room (2,2).
- $HP[2, 1] = 3$, since in room (2,1), Link will first acquire 2 points and needs a total of 5 points when moving on to (2,2).
- $HP[1, 1] = 1$, since if Link starts with 1 point, accumulates 1 point in room (1,1) and another 3 points by entering room (1,2), then Link has a total of 5 points when entering room (2,2), which is just sufficient.
- Link should take the path (1,1), (1,2), (2,2). This is better than the alternative, (1,1), (2,1), (2,2), since in this case Link would need two points initially.

1. For the grid on the right above, what is $H[3, 2]$?

   For the grid on the right, $H[3, 2]$ is 1, since although Link gets 4 points in room (3,2), Link must always have at least one point initially.

2. Next we'll develop a recurrence for $HP[i,j]$. Start with a base case when $i = m$ and $j = n$. What is $H[m,n]$ when $G[m,n]$ is negative? When $G[m,n]$ is nonnegative?

   If $G[m,n]$ is negative, Link needs at least $1 - G[m,n]$ points so that he is still left with one point after paying the cost of $G[m,n]$. If $G[m,n]$ is positive, then one point is sufficient. So we have that

   $$H[m,n] = \begin{cases} 1 - G[m,n], & \text{if } G[m,n] \text{ is negative,} \\ 1, & \text{if } G[m,n] \text{ is nonnegative.} \end{cases}$$

   Equivalently, we can write

   $$H[m,n] = \max\{1, 1 - G[m,n]\}.$$

3. Here is a recurrence for $HP[i,j]$, when $i \le m+1$ and $j \le n+1$. Explain why this is correct.

   $$HP[i,j] = \begin{cases} \text{your solution from part 2} & \text{if } i = m \text{ and } j = n \qquad \text{[base case]} \\ \max(1, \min(HP[i+1,j], HP[i,j+1]) - G[i,j]), & \text{if } 1 \le i \le m-1 \text{ or } 1 \le j \le n-1, \\ \infty, & \text{if } i = m+1 \text{ or } j = n+1 \qquad \text{[boundary cases]} \end{cases}$$

   In the general case, we also ensure that Link has at least one point by taking the max of 1 and another quantity. The additional quantity takes the min over the two possible next steps. Link first accumulates $G[i,j]$ points (either positive or negative), and if he moves downwards, he will need $HP[i,j+1]$ more points to finish the game. So the total points needed is $HP[i,j+1] - G[i,j]$. If Link moves right, we replace $HP[i,j+1]$ by $HP[i+1,j]$ to get $HP[i+1,j] - G[i,j]$. The min of these two options is the best option, unless it is less than 1, in which case $HP[i,j]$ is 1. So we take the max of 1 and $\min(HP[i+1,j], HP[i,j+1]) - G[i,j])$.

   One issue is that if Link reaches the bottom side of the grid or the right side of the grid before getting to room (m,n), one of the two choices is "out of bounds". To handle this, we add an extra case to the recurrence, setting $H[i,j]$ to be ∞ if one or both of i or j is out of bounds.

4. Use the recurrence from the previous two parts (base case plus general case) to write a memoized algorithm to compute $HP[1,1]$.

```
function MemoSaveZelda(G[1...m][1...n])
    HP[1..m][1..n] ← Create array of size m × n
    Initialize all entries to −∞
    Memo-Helper(G[1..m][1..n], 1, 1)
end function

function Memo-Helper(G[1...m][1...n], i, j)
    if HP[i,j] is −∞ then            ▷ solution is not computed yet
        if i == m and j == n then
            HP[i,j] ← max(1, 1 − G[i,j])    ▷ Base case
        else if i == m + 1 or j == n + 1 then
            return ∞             ▷ To handle reaching coordinates outside of the grid
        else
            HP[i,j] ← max{1, min(Memo-Helper(G, i+1, j), Memo-Helper(G, i, j+1))−G[i,j]}
        end if
    end if
    return HP[i,j]
end function
```