CS 520 Final Question 1 Solution

Guo Chen

December 2018

1 Solutions

1.1 Question a)

Solution:

A program is written to count how many white cells (including the one marked as G) are there in the maze. After running the program, I got the number 1239. Since it is equally possible for a bot to stay in any white cell in the maze, therefore the probability that the bot is at G, a certain white cell in the maze, is 1/1239.

1.2 Question b)

Solution:

Initially, the problem appeared quite hard to find a point to break through. However, in order to solve it in an possibly easier way, the whole problem can be restated as following:

Suppose there is one person in each white cell in the maze. Each person can try to move UP, DOWN, LEFT, RIGHT. A person would stay in the same place if blocked by surrounding blocks. Now try to find a sequence of moves that will make at least half of all people in the maze, no matter where their own initial places would be, to move to G.

During this process, an interesting issue is the situation where one person is blocked while his/her surrounding people are trying to come where he/she stands. At this moment, the cell where this "poor" person stands can be regarded as a sinking point where many other persons are gathered. After that, all these people can simply move and try to find their way to G altogether.

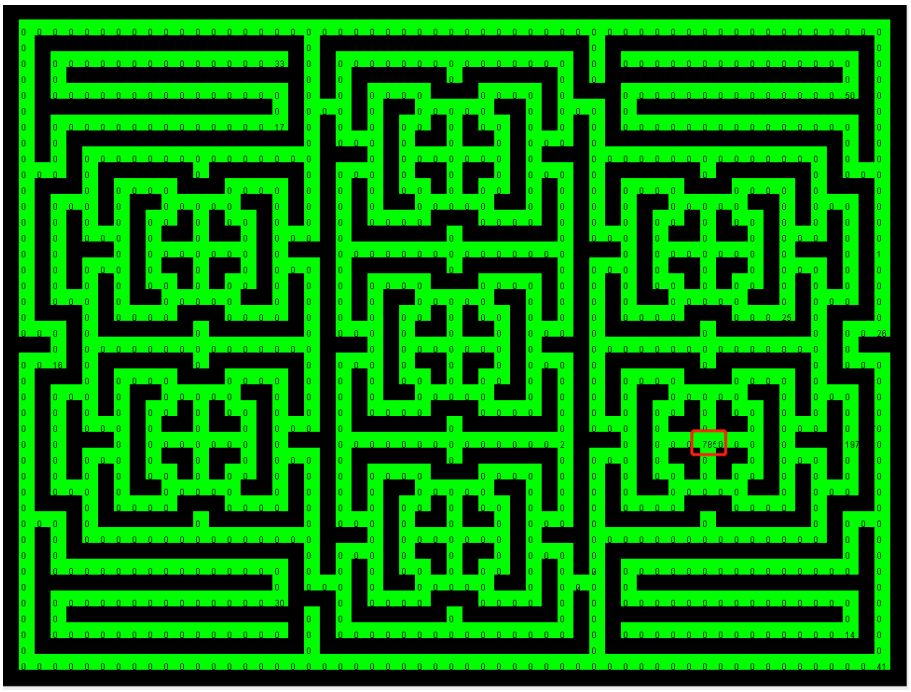
This gives a few inspirations to the solution of the problem. In this restated problem, in order to find out a sequence that can make half of people arrive at G, perhaps it is feasible to divide it into two stages:

1. Guide at least half of all people in the maze to gather at a certain cell (of course, the best case is that all people in the maze can reunion at a certain cell, what a happy thing);

2. Guide these gathered people to cell G.

By catenationg the sequences of movements in these two stages together, a possible sequence that makes at least half of people to cell G can be obtained by far.

Based on such an analysis, I wrote a program, which is attached with this report, to solve the entire problem. By running the program, a sequence of movements that leaves with probability of at least 50% that anyone from any place in the maze could arrive at G is as long as 325, and the specific steps are stored in an attached file named 'b_results.txt'. And its effect is shown as below, where over 780 (more than 620, half of all white cells in the maze) people throughout the maze are guided successfully to G. **(Please note here that all green cells are in fact the white cells as mentioned previously, I used green color here to make my eyes feel more comfortable when looking at it :)**



1.3 Question c)

Solution:

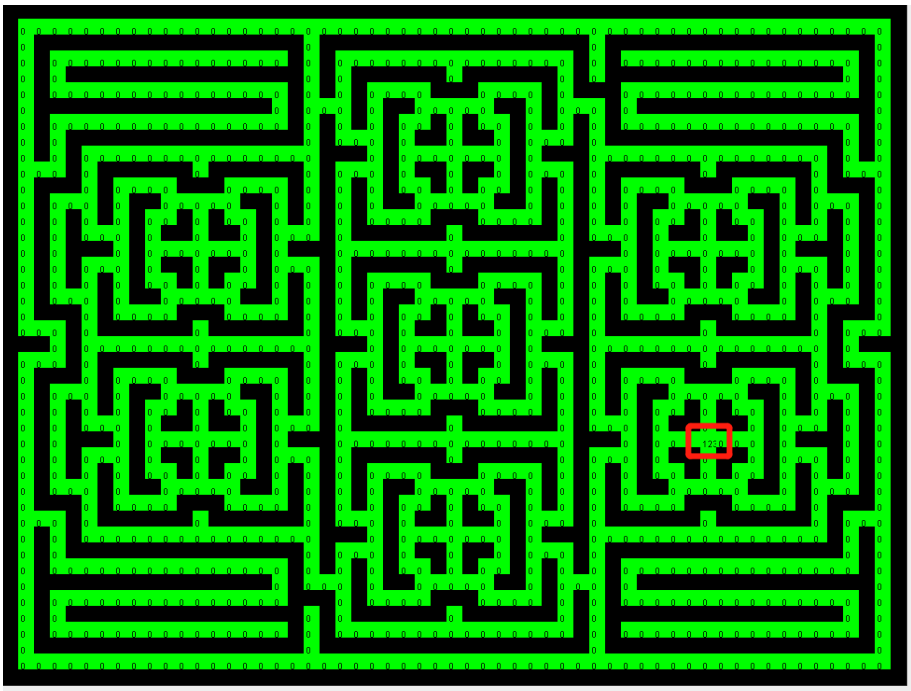
The method to find the shortest possible sequence of moves guarantees that a given bot would arrive at G starting from any cell is quite similar to that in question b) except that the it becomes more challenging this time.

In order to reach the goal, structure of the maze can be made use of to find out the possible shortest path. By taking a close look, there are 7 repeated structures in the maze, which means the same sequence of operations on them

would result in the same (this is very important, because the effect of making any operation here would be multiplied by 7). In addition, since all these structures are surrounded by 4 long aisle, once operations are applied to the structures, the "persons" in these paves would be gathered soon as well since they can only move either in left/right direction or up/down direction. Finally, one may notice that there are four zigzag structures in four corners in the maze. This is the most difficult part. Actually one can take care of it after all "people" from other places are gathered.

What's more, I think there are some practical tricks to avoid as much as possible tedious sequences. One point is to make advantage of the places where movements are restricted to left/right or up/down. For example, if I am to gather some persons in a row, I can keep some groups of gathered "persons" (represented as a number in each cell for visualization) in places where moving left and right is restricted, such as in a vertical aisle surrounded by walls on its left and right sides, so that the gathered "people" would stay in one plac. It would be much more convenient if I am to find them and guide them to G quickly.

By applying these information and techniques, I managed to get all "people" in the maze together using 826 steps. Specific steps are shown in the attached file named c_results.txt. As shown in the picture, currently all "people" are gathered in the cell G. My visualization is not very clear though, the number highlighted with a red rectangle is actually 1239, the total number of all cells in the maze. **(Please note here that all cells colored as green are in fact the white cells as mentioned previously, I used green color here to make my eyes feel more comfortable when looking at it :)**



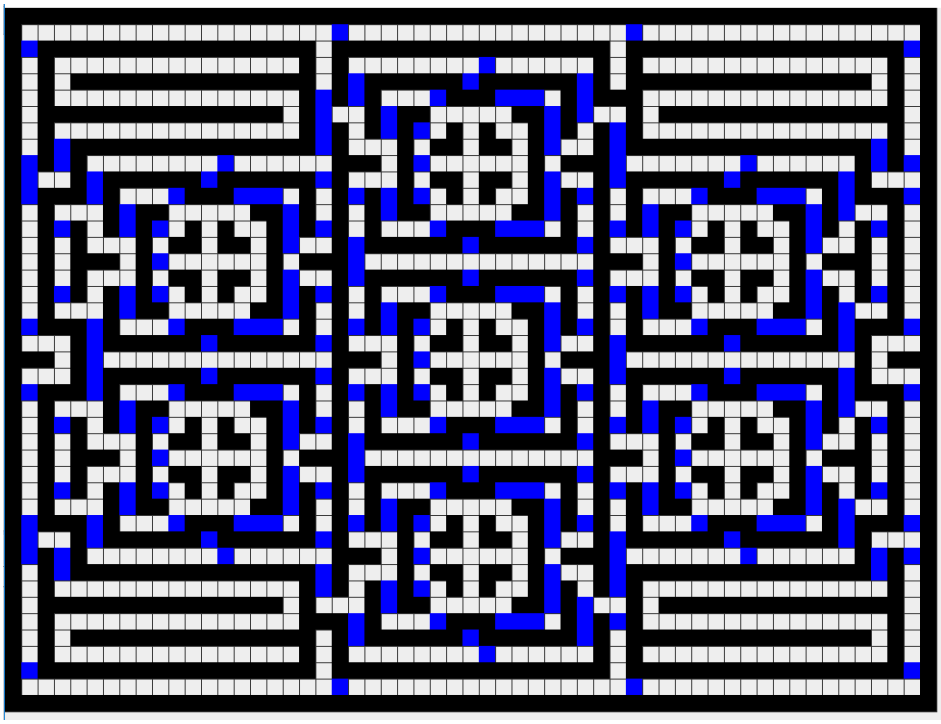
1.4 Question d)

1.4.1 Question d.1)

Solution:

By writing a program, I find out there are 287 white cells where can be regarded as a satisfied starting point of such a movement, therefore the probability that the bot in the cell is 287/1239.

All the satisfying cells are marked blue on the maze map below.



Specifically, there are multiple cases where such a continuous movement can be possible. Intuitively, to satisfy such a description, a bot is merely supposed to move left three times consecutively with the number of its surrounding blocks remain 5 all the time. However, since a bot cannot sense if it hits a wall or not, it is possible that such a series of movement occur to it even if it is stuck in a certain cell, which means it hits the wall 3 times in a row. In addition, it is easy to expand this situation, that is, a bot may find it self in the following cases as well:

1. moves left in two consecutive steps and hits the wall in the third time, when the amount of its surrounding blocks remains 5 all the time;

2. moves left in one step and hits the wall twice afterwards, when the amount of its surrounding blocks remains 5 all the time.

1.4.2 Question d.2)

Solution:

For this part, I will write pseudocode here to describe the algorithm. As for the problem, it gives a series of observations $Y_{0:n}$ together with corresponding actions $A_{0:n-1}$ and asks which cell the bot is most likely to be in at step n. To do inference in such a temporal model, I would like to define the problem as a **filtering** one.

To start with, I will introduce some variables with their notations here and make a brief analysis of how these variables interact to compose the algorithm.

Let C_n be the cell that a bot finds itself in at step n, the probability that the cell is located at C_n can be denoted as $P(C_n|Y_{0:n}, A_{0:n-1})$. Name the probability that the bot is in cell i at step n as $Belief_n(C = i)$. By far the cell where the bot is most likely to be in at step n is exactly the one that makes the belief to reach the maximum value among all others. Now based on the file posted by Prof. Cowan on Sakai, the belief that the bot is in a certain cell i at step n is (assume i' represents a cell the bot possibly stays in step $n-1$):

$Belief_n(i) = P(C_n = i|Y_{0:n}, A_{0:n-1}) = \beta P(C_n = i, Y_n, A_{n-1}|Y_{0:n-1}, A_{0:n-2})$(Based on Bayes' Formula)
$= \beta \sum_{i'} P(C_n = i, C_{n-1} = i', Y_n, A_{n-1}|Y_{0:n-1}, A_{0:n-2})$ (Marginalization)
$= \beta \sum_{i'} P(C_{n-1} = i'|Y_{0:n-1}, A_{0:n-2}) \cdot P(A_{n-1}|C_{n-1} = i', Y_{0:n-1}, A_{0:n-2}) \cdot P(C_n = i|C_{n-1} = i', A_{n-1}, Y_{0:n-1}, A_{0:n-2}) \cdot P(Y_n|C_n = i, C_{n-1} = i', A_{n-1}, Y_{0:n-1}, A_{0:n-2})$.

Based on Bayes' Formula, $P(C_n = i|Y_{0:n}, A_{0:n-1}) = P(C_n = i|Y_n, A_{n-1}, Y_{0:n-1}, A_{0:n-2}) = P(C_n = i, Y_n, A_{n-1}|Y_{0:n-1}, A_{0:n-2})/P(Y_n, A_{n-1}|Y_{0:n-1}, A_{0:n-2})$. Since the sequence of actions $A_{0:n}-1$ and the sequence of observations $Y_{0:n}$ has already been given, therefore, $P(A_{n-1}, Y_n|Y_{0:n-1}, A_{0:n-2}) = 1$, that is, $\beta = 1/P(Y_n, A_{n-1}|Y_{0:n-1}, A_{0:n-2}) = 1$.

Similarly, $P(A_{n-1}|C_{n-1} = i', Y_{0:n-1}, A_{0:n-2}) = 1$.

Besides, the entire model to describe movements of the bot is a Hidden Markov Model. Therefore, based on Markov property as well as given information about the bot and the maze, it is easy to conclude that:

1. The probability that the bot is in cell i at step n, or $P(C_n = i)$, is merely related to where it is in the last step, C_{n-1}, and the action it took in the last step, A_{n-1}.

2. The observation at step n, or Y_n, is related only to which cell the bot is in at step n, C_n, and has nothing to do with either observations or actions multiple steps away, e.g. $Y_{0:n-1}, A_{0:n-1}$.

After combining this information,

$Belief_n(i) = \beta \sum_{i'} P(C_{n-1} = i'|Y_{0:n-1}, A_{0:n-2}) \cdot P(A_{n-1}|C_{n-1} = i', Y_{0:n-1}, A_{0:n-2}) \cdot P(C_n = i|C_{n-1} = i', A_{n-1}, Y_{0:n-1}, A_{0:n-2}) \cdot P(Y_n|C_n = i, C_{n-1} = i', A_{n-1}, Y_{0:n-1}, A_{0:n-2})$
$= \beta \sum_{i'} P(C_{n-1} = i'|Y_{0:n-1}, A_{0:n-2}) \cdot P(C_n = i|C_{n-1} = i', A_{n-1}) \cdot P(Y_n|C_n = i)$.

Based on what has been defined as a belief previously,

$Belief_n(i) = \beta \sum_{i'} Belief_{n-1}(i') \cdot P(C_n = i|C_{n-1} = i', A_{n-1}) \cdot P(Y_n|C_n = i)$,

which has a recursive structure. And based on question a), it is easy to see $Belief_0(i) = 1/1239$ in its base case when $n = 0$.

Therefore, according to the analysis above, I write two algorithms to work together to get the most likely cell (in form of its index).

Algorithm 1 Getting a belief at step k in terms of cell i

procedure BELIEF($k, i, maze, Y_{0:n}, A_{0:n-1}$) ▷ n represents steps so far, i identifies an accessible cell in the maze
    $sum \leftarrow 0$
    $\beta \leftarrow 1$
    if $k = 0$ **then** ▷ Base case
        $value \leftarrow 1/1239$
        return $value$
    else
        if observation Y_k can be obtained at C_{k-1} **then**
            $P(Y_k|C_k = i) \leftarrow 1$
        else
            $P(Y_k|C_k = i) \leftarrow 0$
        for each accessible cell i' in the $maze$ **do**
            if the bot can go to i b from i' by taking action A_{k-1} **then**
                $P(C_k = i|C_{k-1} = i', A_{k-1}) \leftarrow 1$
            else
                $P(C_k = i|C_{k-1} = i', A_{k-1}) \leftarrow 0$
            $sum \leftarrow sum + BELIEF(k-1, i', maze, Y_{0:n}, A_{0:n-1}) \cdot P(C_k = i|C_{k-1} = i', A_{k-1}) \cdot P(Y_k|C_k = i)$ ▷ Use recursive structure
        $value \leftarrow \beta \cdot sum$
        return $value$

Algorithm 2 Finding the best belief at step k

procedure FINDING_BEST_BELIEF($k, maze, Y_{0:n}, A_{0:n-1}$) ▷ n represents steps so far
    $max\_index = -1$ ▷ Record which cell is the most likely one
    $max\_value = 0$ ▷ Record the maximum probability that the bot is in cell max_index
    for each accessible cell i in the maze **do**
        if $max\_value < BELIEF(k, i, maze, Y_{0:n}, A_{0:n-1})$ **then**
            $max\_index \leftarrow i$
            $max\_value \leftarrow BELIEF(k, i, maze, Y_{0:n}, A_{0:n-1})$
    return max_index