DAA ASSIGNMENT 5
Group No 20

Harsh Mahajan (IIB2019001)

Pradhuman Singh Baid (IIB2019002)

Vasu Gupta(IIB2019003)

Contents -

Problem Statement

- Given an NxN chessboard and a Knight at position (x,y).The Knight has to take exactly K steps, where at each step it chooses any of the 8 directions uniformly at random. What is the probability that the Knight remains in the chessboard after taking K steps, with the condition that it can’t enter the board again once it leaves it? Solve using Dynamic programming

Introduction

Dynamic Programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler sub-problems and utilizing the fact that the optimal solution to the overall problem depends upon the optimal solution to its sub-problems.

ALGORITHMIC DESIGN

One thing that we can observe is that at every step the Knight has 8 choices to choose from. Suppose, the Knight has to take k steps and after taking the Kth step the knight reaches (x,y). There are 8 different positions from where the Knight can reach to (x,y) in one step, and they are: (x+1,y+2), (x+2,y+1), (x+2,y-1), (x+1,y2), (x-1,y-2), (x-2,y-1), (x-2,y+1), (x-1,y+2).

The final probability after K steps will simply be equal to the (probability of reaching each of these 8 positions after K-1 steps)/8.

We are dividing by 8 because each of these 8 positions has 8 choices and position (x,y) is one of the choices.

For the positions that lie outside the board, we will either take their probabilities as 0 or simply neglect it.

Since we need to keep track of the probabilities at each position for every number of steps, we need Dynamic Programming to solve this problem.

Time complexity

In this method, we are working on n × n elements and there are k layers considered

Therefore, Time complexity would be O(n^2× k)

Space complexity

The size of dp array is $n \times n$ and some constant variables are used.

Therefore, Space complexity is $O(n^2)$

Conclusion

Solutions in which some steps are repeating again and again can be made efficient using dynamic programming

References

- [1] Wikipedia: Dynamic Programming,
  https://en.wikipedia.org/wikiDynamic_programming
- [2] GeeksforGeeks: Dynamic Programming,
  https://www.geeksforgeeks.org/dynamic-programming