CS5120: Probability in Computing
Assignment 4

Sagar Jain
CS17BTECH11034

Solutions

1. The probability distribution vector after four time steps is $[0.1938, 0.1772, 0.629]$.

2. The expected time from i to j is $j^2 - i^2$.

3. The expected number of steps to reach (1, 1, 1) from (0, 0, 0) is 10.

4. The stationary distribution for a random walk on G is $[2/7, 1/7, 3/14, 1/7, 1/7, 1/14]$.

5. (a) We give a constructive proof to show that there exists a coloring of the graph G with 2 colors such that no triangle is monochromatic. Since there exists a valid 3 colouring, let the 3 colours used to colour the graph G be, C1, C2 & C3. Colour all the nodes that are coloured C1 to C2. We claim that there are no monochromatic triangles in the so formed graph.
   It is easy to see that in the orignal colouring every traingle contains all the 3 colours (any less and it would not be a valid 3 colouring). Now since we have only changed C1 to C2 it follows that every graph contains 2 colours. Hence Proved.

   (b) **The Algorithm**

   - Initialise an empty 3-SAT formula.
   - For every 3 vertices, check if they form a triangle, if they do, add these two clauses to the formula $(x_i \vee x_j \vee x_k)$, $(\bar{x_i} \vee \bar{x_j} \vee \bar{x_k})$. Here the variable x_i is true if node i is coloured black and is false if node i is coloured white.
   - Solve the 3-SAT using Schoning's algorithm for 3-SAT

   First we note that if the 3-SAT formula can be satisfied then it means that for every triangle in the graph both the clauses are satisfied. This would mean that for every triangle we have atleast one white node and one black node, this means that no triangle is monochromatic. A solution always exists since the graph is 3-colourable(follows from previous bit).
   Analysis
   The time complexity of step 1 is $O(1)$, the time complexity of step 2 is $O(n^3)$ since there can be at most C_3^n triangles in the graph. As a result of this the number of clauses in the 3-SAT formula are $O(n^3)$. The complexity of step 3 is $O^*(1.33^{n^3})$.

Therefore, the overall complexity of the algorithm is $O^*(1.33^{n^3})$. It should be noted that a non probabilistic brute force algoritm would have a complexity of $O(2^n)$.