**POSTLAB 8**

**1) Give the efficient approach to handle M and write reason**

**Ans: -**

->To compute the PageRank for a large graph representing the Web, we have to perform a matrix–vector multiplication on the order of 50 times, until the vector is close to unchanged at one iteration.

->The transition matrix of the Web M is very sparse. Thus, representing it by all its elements is highly inefficient. Rather, we want to represent the matrix by its nonzero elements.

->The transition matrix is very sparse, since the average Web page has about 10 out-links. If, say, we are analysing a graph of ten billion pages, then only one in a billion entries is not 0. The proper way to represent any sparse matrix is to list the locations of the nonzero entries and their values. If we use 4-byte integers for coordinates of an element and an 8-byte double-precision number for the value, then we need 16 bytes per nonzero entry. That is, the space needed is linear in the number of nonzero entries, rather than quadratic in the side of the matrix.

->However, for a transition matrix of the Web, there is one further compression that we can do. If we list the nonzero entries by column, then we know what each nonzero entry is; it is 1 divided by the out-degree of the page. We can thus represent a column by one integer for the out-degree, and one integer per nonzero entry in that column, giving the row number where that entry is located. Thus, we need slightly more than 4 bytes per nonzero entry to represent a transition matrix.

M = [ 0 1/2 1 0,

1/3 0 0 1/2,

1/3 0 0 1/2,

1/3 1/2 0 0 ]

Source Degree Destinations

A 3 B, C, D

B 2 A, D

C 1 A

D 2 B, C

->For instance, the entry for A has degree 3 and a list of three successors. We can deduce that the column for A in matrix M has 0 in the row for A (since it is not on the list of destinations) and 1/3 in

the rows for B, C, and D. We know that the value is 1/3 because the degree column tells us there are three links out of A.

**2) Give improvement of Page rank algorithm for spider trap problem and dead end**

**Ans: -**

->A spider trap is a set of nodes that, while they may link to each other, have no links out to other nodes. In an iterative calculation of PageRank, the presence of spider traps causes all the PageRank to be captured within that set of nodes.

->Groups of pages that all have outlines but they never link to any other pages are called spider raps.

->The problem is solved by a method called “taxation,” where we assume a random surfer has a finite probability of leaving the Web at any step, and new surfers are started at each page.

->These structures can appear intentionally or unintentionally on the Web, and they cause the PageRank calculation to place all the PageRank within the spider traps.

->To avoid the problem, we modify the calculation of PageRank by allowing each random surfer a small probability of teleporting to a random page, rather than following an out-link from their current page.

->The iterative step, where we compute a new vector estimate of PageRank v′ from the current PageRank estimate v and the transition matrix M is v′ = βMv + (1 − β)e/n where β is a chosen constant, usually in the range 0.8 to 0.9, e is a vector of all 1’s with the appropriate number of components, and n is the number of nodes in the Web graph. The term βMv represents the case where, with probability β, the random surfer decides to follow an out-link from their present page. The term (1 − β) e/n is a vector each of whose components has value (1 − β)/n and represents the introduction, with probability 1 − β, of a new random surfer at a random page. Note that if the graph has no dead ends, then the probability of introducing a new random surfer is exactly equal to the probability that the random surfer will decide not to follow a link from their current page. In this case, it is reasonable to visualize the surfer as deciding either to follow a link or teleport to a random page. However, if there are dead ends, then there is a third possibility, which is that the surfer goes nowhere. Since the term (1 − β)e/n does not depend on the sum of the components of the vector v, there will always be some fraction of a surfer operating on the Web. That is, when there are dead ends, the sum of the components of v may be less than 1, but it will never reach 0.