Pseudo code

① The program uses the graph class to create links and uses a couple of functions to calculate Ranks and store.

① Calculate Rank:

```
create_table 2d;
first_row ← (1/n)
for (i=0, i<3; i++){
    for(j=0 ;j<web.size(), j++){
        var1 = numof ingoingedges
        var = get incoming edges        } Graph class
        for(m=0, m<var.size(), m++){

            rank = rank + (table[i-1][var[j]]/
                           numofoutgoing.
        }
    nthrank ← rank
    }
}
return . finaltable[2]
```

(2) SoRt rank:

Uses Bubble soRt algorithm to Sort the veetor aecording to initial Ranking (Ascendingly)

(3) Sort_Score:

Use Bubble sort algorithm to Sort the vecdor acording to the Score (Descendingly)

Space and time complexity:

Complexity = Complexity of (calculate_rank)

+ Complexity of (sortRank)

+ Complexity of (sort_Score)

① Space complexity: Number of Accessing array elements
↳ {table sizes}

- final table has $n \times n = n^2$ elements

  where n is the number of Nodes in the webpage graph.

- Number of accessing array elements:

$$T(n) = \sum_{i=0}^{3} \sum_{j=0}^{n} \sum_{m=0}^{k} 2 = \sum_{i=0}^{3} \sum_{j=0}^{n} 2(k+1)$$

$$= \sum_{i=0}^{3} 2(n+1)(k+1)$$

$$= 8(n+1)(k+1)$$

k = num of outgoing edges

$$T(n) = O(kn)$$

$$T(n) = kn + \underbrace{n^2 + n^2}_{\text{Bubble sort}}$$

$$= O(n^2)$$

- Number of Arithmetic operations:

$$T(n)_{temp} = \sum_{i=0}^{3} \sum_{j=0}^{n} \sum_{m=0}^{K} 2 = O(Kn)$$

$$T(n) = Kn + 2n^2 \qquad T(n) = O(n^2)$$

(3-) Main Data structures are Vectors and Graphs.