Program 11: Quick Sort

Program:

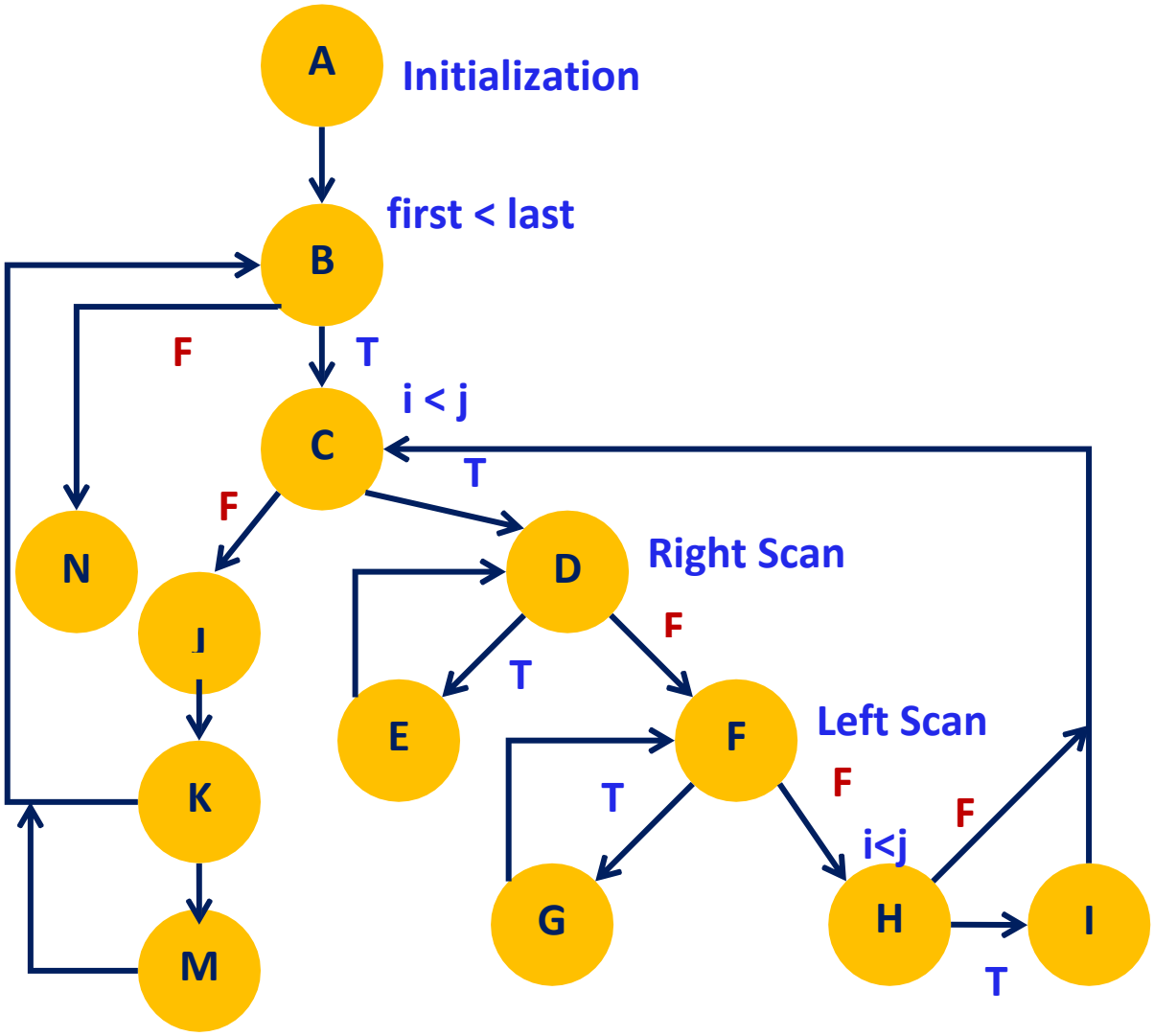
```c
#include<stdio.h>
void quicksort(int x[10],int first,int last)
{
    int temp,pivot,i,j;
    if(first<last)
    {
        pivot=first;
        i=first;
        j=last;
        while(i<j)
        {
            while(x[i]<=x[pivot] && i<last)
            i++;
            while(x[j]>=x[pivot])
            j--;
            if(i<j)
            {
                temp=x[i];
                x[i]=x[j];
                x[j]=temp;
            }
        }
        temp=x[pivot];
        x[pivot]=x[j];
        x[j]=temp;
        quicksort(x,first,j-1);
        quicksort(x,j+1,last);
    }
}
int main()
{
    int a[20],i,key,n;
    printf("Enter the size of the array: ");
    scanf("%d",&n);
    if(n>0)
    {
        printf("Enter the elements of the array: ");
        for(i=0;i<n;i++)
        scanf("%d",&a[i]);
```

```
        quicksort(a,0,n-1);
        printf("The elements in the sorted array is:\n");
        for(i=0;i<n;i++)
            printf("%d\t",a[i]);
    }
    else{
        printf("Size of array is invalid\n");
    }
}
```

Output:

```
sooraj@Asus-F-15:~/st-lab$ ./a.out
Enter the size of the array: 5
Enter the elements of the array: 5
1
4
3
1
The elements in the sorted array is: 1 2 3 4 5
```

Program graph:



Independent Paths– Quick Sort

P1: A-B-N
P2: A-B-C-J-K-B
P3: A-B-C-J-K-M-B
P4: A-B-C-D-F-H-C
P5: A-B-C-D-F-H-I-C
P6: A-B-C-D-E-D-F-H
P7: A-B-C-D-F-G-F-H

Independent Paths: #Edges=18, #Nodes=13, #P=1 **V(G)**= E-N+2P = 18-13+2 = **7**

Test Cases:

Paths	Inputs		Expected Output	Remarks
	X[]	First , Last		
P1: A-B-N	5	1,1	Sorted	Only one Elem
P2: A-B-C-J-K-B	5,4	1,2	Repeat & Sorted	Two Elements
P3: A-B-C-J-K-M-B	1,2,3 OR 3,1,2	1,3	Repeat & Sorted	Three Elements
P4: A-B-C-D-F-H-C	1,2,3,4,5	1,5	Repeat & Sorted	ASC Sequence
P5: A-B-C-D-F-H-I-C	5,4,3,2,1	1,5	Repeat & Sorted	DSC Sequence
P6: A-B-C-D-E-D-F-H	1,4,3,2,5 OR 2,2,2,2,2	1,5	Repeat & Sorted	Pivot is MIN
P7: A-B-C-D-F-G-F-H	5,2,3,1,4	1,5	Repeat & Sorted	Pivot is MAX