**Assignment - 2**

1. Write a program of Bubble sort and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <sys/time.h>

# include <random>

void bubbleSort(int [],int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    bubbleSort(arr,n);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %ld",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

void bubbleSort(int arr[],int n)

{

    int exchg;

    for(int i=0; i<n; i++)

    {

        exchg=0;

        for(int j=0; j<n-i; j++)

        {

            if(arr[j]>arr[j+1])

            {

                int t=arr[j];

                arr[j]=arr[j+1];

                arr[j+1]=t;

                exchg++;

            }

        }

        if(exchg==0)

        {

            break;

        }

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 15

Enter the number of elements: 100

Time Difference : 17

Enter the number of elements: 100

Time Difference : 13

Average Case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 27

Worst Case :-

Enter the number of elements: 100

Time Difference : 25

Enter the number of elements: 100

Time Difference : 24

Enter the number of elements: 100

Time Difference : 24

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 15 | 17 | 13 | 27 | 33 | 29 | 25 | 24 | 24 |
| 200 | 55 | 56 | 66 | 111 | 104 | 98 | 113 | 120 | 102 |
| 300 | 144 | 124 | 108 | 214 | 237 | 212 | 217 | 270 | 269 |

* Here we can see that for size of array equal to 100 time taken by best case is least.
* Also that time taken by worst case is less than average case.
* For array size 200 time taken is least in best case then average case then worst case.
* For array size 300 time taken is least in best case then average case then worst case.
* Also that time taken increases with increase in size of array.
* And time difference between best case, average case and worst case also increases with increase in size of array.
* Here we can also observe that time difference in average case and worst case is not so much.

Conclusion :-

Here we can say Big O of best case of bubble sort is O(n).

Big O of average case and worst case is O(n2).

2. Write a program of Selection and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <sys/time.h>

# include <random>

void selectionSort(int [],int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    selectionSort(arr,n);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %ld",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

void selectionSort(int arr[],int n)

{

    for(int pass=0; pass<n-1; pass++)

    {

        int min\_index = pass;

        for(int i=pass+1; i<n; i++)

        {

            if(arr[i]<arr[min\_index])

            {

                min\_index=i;

            }

        }

        if(min\_index!=pass)

        {

            int temp=arr[pass];

            arr[pass]=arr[min\_index];

            arr[min\_index]=temp;

        }

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 13

Enter the number of elements: 100

Time Difference : 19

Enter the number of elements: 100

Time Difference : 16

Average case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 25

Worst case :-

Enter the number of elements: 100

Time Difference : 13

Enter the number of elements: 100

Time Difference : 13

Enter the number of elements: 100

Time Difference : 16

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 13 | 19 | 16 | 25 | 19 | 18 | 13 | 13 | 16 |
| 200 | 58 | 57 | 53 | 74 | 80 | 65 | 41 | 46 | 43 |
| 300 | 115 | 126 | 113 | 162 | 121 | 136 | 113 | 149 | 103 |

* Here we can see that for size of array equal to 100 time taken by worst case is least.
* Also that time taken by best case and worst case is less than average case.
* For array size 200 time taken is least in worst case then best case then average case.
* For array size 300 time taken is least in best case then worst case then average case.
* Also that time taken increases with increase in size of array.
* And time difference between best case, average case and worst case doesn’t much increases with increase in size of array.
* Here we can also observe that time difference in best case, average case and worst case is not so much.

Conclusion :-

Here we can say that Big O of best case, average case and worst case is O(n2).

3. Write a program of Quick sort and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <sys/time.h>

# include <random>

void quickSort(int [],int,int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    quickSort(arr,0,n-1);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %d",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

void quickSort(int arr[],int lb,int ub)

{

    if(lb<ub)

    {

        int key=arr[lb];

        int left=lb;

        int right=ub+1;

        while(1)

        {

            left++;

            while(key>arr[left])

            {

                left++;

            }

            right--;

            while(key<arr[right])

            {

                right--;

            }

            if(left<right)

            {

                int temp = arr[left];

                arr[left] = arr[right];

                arr[right] = temp;

            }

            else

            {

                break;

            }

        }

        arr[lb]=arr[right];

        arr[right]=key;

        quickSort(arr,lb,right-1);

        quickSort(arr,right+1,ub);

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 9

Enter the number of elements: 100

Time Difference : 9

Enter the number of elements: 100

Time Difference : 12

Average case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 7

Worst case :-

Enter the number of elements: 100

Time Difference : 10

Enter the number of elements: 100

Time Difference : 10

Enter the number of elements: 100

Time Difference : 13

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 9 | 9 | 12 | 7 | 6 | 5 | 10 | 10 | 13 |
| 200 | 42 | 45 | 37 | 13 | 14 | 12 | 35 | 37 | 37 |
| 300 | 110 | 111 | 114 | 21 | 24 | 20 | 77 | 68 | 51 |

* Here we can see that for size of array equal to 100 time taken by average case is least.
* Also that time taken by best case and average case is less than worst case.
* For array size 200 time taken is least in average case then best case then worst case.
* For array size 300 time taken is least in average case then worst case then best case.
* Also that time taken increases with increase in size of array.
* And time difference between best case, average case and worst case doesn’t increase much with increase in size of array.
* Here we can also observe that time difference in best case, average case and worst case is not so much when number of elements to sort are low.
* As we increase number of elements in array time difference in average case, best case and worst case increases.

Conclusion :-

Here we can say that Big O of best case, average case and worst case is O(n\*log(n)).

4. Write a program of Merge sort and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <stdlib.h>

# include <sys/time.h>

# include <random>

void merge(int[],int,int,int);

void split(int[],int,int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    split(arr,0,n-1);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %d",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

void merge(int arr[],int first,int second,int third)

{

    int temp[third-first+1];

    int i=first,j=second,k=0;

    while(i<second && j<=third)

    {

        if(arr[i]<arr[j])

        {

            temp[k]=arr[i];

            i++;

        }

        else

        {

            temp[k]=arr[j];

            j++;

        }

        k++;

    }

    if(i==second)

    {

        while(j<=third)

        {

            temp[k]=arr[j];

            k++;

            j++;

        }

    }

    else

    {

        while(i<second)

        {

            temp[k]=arr[i];

            i++;

            k++;

        }

    }

    for(int l=0; l<=third-first; l++)

    {

        arr[first+l]=temp[l];

    }

}

void split(int arr[],int start,int finish)

{

    if(finish>start)

    {

        int middle = (start+finish+1)/2;

        split(arr,start,middle-1);

        split(arr,middle,finish);

        merge(arr,start,middle,finish);

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 6

Enter the number of elements: 100

Time Difference : 9

Enter the number of elements: 100

Time Difference : 7

Average case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 13

Worst case :-

Enter the number of elements: 100

Time Difference : 8

Enter the number of elements: 100

Time Difference : 10

Enter the number of elements: 100

Time Difference : 11

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 6 | 9 | 7 | 13 | 14 | 10 | 8 | 10 | 11 |
| 200 | 21 | 24 | 26 | 34 | 35 | 29 | 17 | 16 | 15 |
| 300 | 31 | 24 | 33 | 59 | 43 | 37 | 24 | 22 | 33 |

* Here we can see that for size of array equal to 100 time taken by best case is least.
* Also that time taken by best case and worst case is less than average case.
* For array size 200 time taken is least in worst case then best case then average case.
* For array size 300 time taken is least in worst case then best case then average case.
* Also that time taken increases with increase in size of array.
* And time difference between best case, average case and worst case doesn’t increase much with increase in size of array.
* Here we can also observe that time difference in best case, average case and worst case is not so much when number of elements to sort are low.
* As we increase number of elements in array time difference in average case increases but time difference in best case and worst case remains almost constant.

Conclusion :-

Here we can say that Big O of best case, average case and worst case is O(n\*log(n)).

5. Write a program of Radix sort and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <sys/time.h>

# include <random>

int findMax(int[],int);

int maxPass(int);

void radixSort(int [],int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    radixSort(arr,n);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %d",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

int findMax(int arr[],int n)

{

    int max = arr[0];

    for(int i=0; i<n; i++)

    {

        if(arr[i]>max)

        {

            max=arr[i];

        }

    }

    return max;

}

int maxPass(int max)

{

    int r=0;

    while(max!=0)

    {

        max=max/10;

        r++;

    }

    return r;

}

void radixSort(int arr[],int n)

{

    int pass = maxPass(findMax(arr,n));

    int bucket\_pass[10][n];

    int r,divisor=1;

    for(int ps=0; ps<pass; ps++)

    {

        int bucket[10]={0};

        for(int i=0; i<n; i++)

        {

            r=(arr[i]/divisor)%10;

            bucket\_pass[r][bucket[r]]=arr[i];

            bucket[r]++;

        }

        int k=0;

        for(int i=0; i<10; i++)

        {

            for(int j=0; j<bucket[i] ;j++)

            {

                arr[k] = bucket\_pass[i][j];

                k++;

            }

        }

        divisor\*=10;

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 9

Enter the number of elements: 100

Time Difference : 6

Enter the number of elements: 100

Time Difference : 7

Average case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 5

Worst case :-

Enter the number of elements: 100

Time Difference : 6

Enter the number of elements: 100

Time Difference : 9

Enter the number of elements: 100

Time Difference : 9

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 9 | 6 | 7 | 5 | 5 | 5 | 6 | 9 | 9 |
| 200 | 12 | 16 | 13 | 15 | 14 | 15 | 12 | 16 | 18 |
| 300 | 18 | 20 | 19 | 20 | 27 | 21 | 19 | 20 | 24 |

* Here we can see that for size of array equal to 100 time taken by average case is least.
* Also that time taken by best case and worst case is greater than average case.
* For array size 200 time taken in worst case, best case and average case is almost constant.
* For array size 300 time taken in worst case, best case and average case is almost constant.
* Also that time taken increases with increase in size of array.
* And time difference between best case, average case and worst case doesn’t increase much with increase in size of array.
* Here we can also observe that time difference in best case, average case and worst case is not so much when number of elements to sort are high.

Conclusion :-

Here we can say that Big O of best case, average case and worst case is O(n\*d).

6. Write a program of Insertion sort and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <sys/time.h>

# include <random>

void insertionSort(int[],int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    insertionSort(arr,n);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %d",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

void insertionSort(int arr[],int size)

{

    for(int i=0 ; i<size; i++)

    {

        int key=arr[i];

        int j;

        for(j=i-1; j>=0; j--)

        {

            if(key<arr[j])

            {

                arr[j+1]=arr[j];

            }

            else

            {

                break;

            }

        }

        arr[j+1]=key;

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 1

Enter the number of elements: 100

Time Difference : 1

Enter the number of elements: 100

Time Difference : 2

Average case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 14

Worst case :-

Enter the number of elements: 100

Time Difference : 15

Enter the number of elements: 100

Time Difference : 14

Enter the number of elements: 100

Time Difference : 21

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 1 | 1 | 2 | 14 | 9 | 14 | 15 | 14 | 21 |
| 200 | 1 | 1 | 1 | 49 | 34 | 40 | 69 | 61 | 74 |
| 300 | 2 | 1 | 1 | 110 | 76 | 87 | 178 | 181 | 206 |

* Here we can see that for size of array equal to 100 time taken by best case is least.
* Also that time taken by best case and average case is less than worst case.
* For array size 200 time taken is least in best case then average case then worst case.
* For array size 300 time taken is least in best case then average case then worst case.
* Also that time taken increases with increase in size of array except for best case.
* And time difference between average case and worst case doesn’t increase much with increase in size of array.
* Here we can also observe that time difference in best case does not change with increase in size of array.

Conclusion :-

Here we can say Big O of best case of bubble sort is O(n).

Big O of average case and worst case is O(n2).

7. Write a program of Merge sort and do the time analysis depending of its

Best case, average case and worst case.

# include <stdio.h>

# include <sys/time.h>

# include <random>

void createHeap(int[],int);

void HeapSort(int[],int);

int main()

{

    int n;

    struct timeval v1,v2;

    struct timezone z1,z2;

    printf("\nEnter the number of elements: ");

    scanf("%d",&n);

    int arr[n];

    for(int i=0; i<n; i++) // Best Case

    {

        arr[i] = i+1;

    }

    for(int i=0; i<n; i++) // Average Case

    {

        arr[i] = rand()%n;

        printf("Info of arr[%d]: %d\n",i+1,arr[i]);

    }

    for(int i=0; i<n; i++) // Worst Case

    {

        arr[i] = n-i;

    }

    gettimeofday(&v1,&z1);

    HeapSort(arr,n);

    gettimeofday(&v2,&z2);

    for(int i=0; i<n; i++) //display loop

    {

        printf("\narr[%d]: %d",i+1,arr[i]);

    }

    printf("\nTime Difference : %ld",(v2.tv\_usec-v1.tv\_usec));

    return 0;

}

void createHeap(int arr[],int n)

{

    int q=1;

    int key,i,j;

    while(q<n)

    {

        i=q;

        key=arr[i];

        j=i/2;

        while(i>0 && key>arr[j])

        {

            arr[i]=arr[j];

            i=j;

            j=i/2;

        }

        arr[i]=key;

        q++;

    }

}

void HeapSort(int arr[],int n)

{

    createHeap(arr,n);

    int i,j,key,q=n-1;

    while(q>0)

    {

        int temp=arr[0];

        arr[0]=arr[q];

        arr[q]=temp;

        i=0;

        key=arr[0];

        j=1;

        while(j+1<q && arr[j+1]>arr[j])

        {

            j++;

        }

        while(j<q && key<arr[j])

        {

            arr[i]=arr[j];

            i=j;

            j=2\*i+1;

            if(j+1<q && arr[j+1]>arr[j])

            {

                j++;

            }

            else if(j>n-1)

            {

                j=n-1;

            }

            arr[i]=key;

        }

        q--;

    }

}

Output :-

Best case :-

Enter the number of elements: 100

Time Difference : 13

Enter the number of elements: 100

Time Difference : 12

Enter the number of elements: 100

Time Difference : 11

Average case :-

Enter the number of elements: 100

Info of arr[1]: 83

Info of arr[2]: 86

Info of arr[3]: 77

Info of arr[4]: 15

Info of arr[5]: 93

Info of arr[6]: 35

Info of arr[7]: 86

Info of arr[8]: 92

Info of arr[9]: 49

Info of arr[10]: 21

Info of arr[11]: 62

Info of arr[12]: 27

Info of arr[13]: 90

Info of arr[14]: 59

Info of arr[15]: 63

Info of arr[16]: 26

Info of arr[17]: 40

Info of arr[18]: 26

Info of arr[19]: 72

Info of arr[20]: 36

Info of arr[21]: 11

Info of arr[22]: 68

Info of arr[23]: 67

Info of arr[24]: 29

Info of arr[25]: 82

Info of arr[26]: 30

Info of arr[27]: 62

Info of arr[28]: 23

Info of arr[29]: 67

Info of arr[30]: 35

Info of arr[31]: 29

Info of arr[32]: 2

Info of arr[33]: 22

Info of arr[34]: 58

Info of arr[35]: 69

Info of arr[36]: 67

Info of arr[37]: 93

Info of arr[38]: 56

Info of arr[39]: 11

Info of arr[40]: 42

Info of arr[41]: 29

Info of arr[42]: 73

Info of arr[43]: 21

Info of arr[44]: 19

Info of arr[45]: 84

Info of arr[46]: 37

Info of arr[47]: 98

Info of arr[48]: 24

Info of arr[49]: 15

Info of arr[50]: 70

Info of arr[51]: 13

Info of arr[52]: 26

Info of arr[53]: 91

Info of arr[54]: 80

Info of arr[55]: 56

Info of arr[56]: 73

Info of arr[57]: 62

Info of arr[58]: 70

Info of arr[59]: 96

Info of arr[60]: 81

Info of arr[61]: 5

Info of arr[62]: 25

Info of arr[63]: 84

Info of arr[64]: 27

Info of arr[65]: 36

Info of arr[66]: 5

Info of arr[67]: 46

Info of arr[68]: 29

Info of arr[69]: 13

Info of arr[70]: 57

Info of arr[71]: 24

Info of arr[72]: 95

Info of arr[73]: 82

Info of arr[74]: 45

Info of arr[75]: 14

Info of arr[76]: 67

Info of arr[77]: 34

Info of arr[78]: 64

Info of arr[79]: 43

Info of arr[80]: 50

Info of arr[81]: 87

Info of arr[82]: 8

Info of arr[83]: 76

Info of arr[84]: 78

Info of arr[85]: 88

Info of arr[86]: 84

Info of arr[87]: 3

Info of arr[88]: 51

Info of arr[89]: 54

Info of arr[90]: 99

Info of arr[91]: 32

Info of arr[92]: 60

Info of arr[93]: 76

Info of arr[94]: 68

Info of arr[95]: 39

Info of arr[96]: 12

Info of arr[97]: 26

Info of arr[98]: 86

Info of arr[99]: 94

Info of arr[100]: 39

Time Difference : 10

Worst case :-

Enter the number of elements: 100

Time Difference : 9

Enter the number of elements: 100

Time Difference : 11

Enter the number of elements: 100

Time Difference : 11

Analysis :-

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Array | Best  Case | | | Average  Case | | | Worst  Case | | |
| 100 | 13 | 12 | 11 | 10 | 11 | 13 | 9 | 11 | 11 |
| 200 | 24 | 28 | 27 | 26 | 32 | 24 | 22 | 21 | 17 |
| 300 | 67 | 53 | 52 | 36 | 38 | 40 | 32 | 34 | 29 |

* Here we can see that for size of array equal to 100 time taken by worst case is least.
* Also that time taken by worst case and average case is less than best case.
* For array size 200 time taken is least in worst case then average case then best case.
* For array size 300 time taken is least in worst case then average case then best case.
* Also that time taken increases with increase in size of array.
* And time difference between average case and worst case doesn’t increase much with increase in size of array.
* Here we can also observe that time difference in best case becomes larger with respect to average case and worst case change with increase in size of array.

Conclusion :-

Here we can say that Big O of best case, average case and worst case is O(n\*log(n)).