| **Name** | **Ameya Dabholkar** |
| --- | --- |
| **UID** | **2021300023** |
| **Subject** | **Data Analysis Algorithm** |
| **Experiment No** | **1** |

**Aim-**

1. To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.
2. Experiment on finding the running time of an algorithm.

**Algorithm-**

1. **Insertion sort-**
   1. procedure insertionSort(A: list of sortable items)
   2. n = length(A)
   3. for i = 1 to n - 1 do
   4. j = i
   5. while j > 0 and A[j-1] > A[j] do
   6. swap(A[j], A[j-1])
   7. j = j - 1
   8. end while
   9. end for
   10. end procedure
2. **Selection sort-**
   1. Repeat Steps b and c for i = 0 to n-1
   2. CALL SMALLEST(arr, i, n, pos)
   3. SWAP arr[i] with arr[pos]
   4. [END OF LOOP]
   5. EXIT

* 1. SMALLEST (arr, i, n, pos)
  2. [INITIALIZE] SET SMALL = arr[i]
  3. [INITIALIZE] SET pos = i
  4. Repeat for j = i+1 to n
  5. if (SMALL > arr[j])
  6. SET SMALL = arr[j]
  7. SET pos = j
  8. [END OF if]
  9. [END OF LOOP]
  10. RETURN pos

**Code-**

**1.A-**

1. **#include <stdio.h>**
2. **#include <stdlib.h>**
3. **#include <math.h>**
4. **long fact(int num){**
5. **if(num == 0){**
6. **return num;**
7. **}**
8. **else{**
9. **return num\*fact(num-1);**
10. **}**
11. **}**
12. **void main(){**
13. **printf("Function 1: n cube\n");**
14. **for(double i=0;i<=100;i++){**
15. **printf("%.0lf\n",pow(i,3));**
16. **}**
17. **printf("Function 2: 2 raised to root 2 log n\n");**
18. **for(double i = 0;i<=100;i++){**
19. **printf("%.3lf\n", pow(2, sqrt(2\*log2(i))));**
20. **}**
21. **printf("Function 3: (3/2) raised to n\n");**
22. **for(double i = 0;i<=100;i++) {**
23. **printf("%.3lf\n",pow((3.0/2.0),i));**
24. **}**
25. **printf("Function 4: n log n\n");**
26. **for(double i = 0;i<=100;i++) {**
27. **printf("%.3lf\n",i \* log2(i));**
28. **}**
29. **printf("Function 5: ln n\n");**
30. **for(double i = 0;i<=100;i++) {**
31. **printf("%.3lf\n",log(i));**
32. **}**
33. **printf("Function 6: 2 raised to 2 raised to n\n");**
34. **for(double i = 0;i<=100;i++) {**
35. **printf("%.3lf\n",pow(2,pow(2,i)));**
36. **}**
37. **printf("Function 7: n\n");**
38. **for(double i = 0;i<=100;i++) {**
39. **printf("%.3lf\n",i);**
40. **}**
41. **printf("Function 8: 2 raised to 2 raised to n+1\n");**
42. **for(double i = 0;i<=100;i++) {**
43. **printf("%.3lf\n",pow(2,pow(2,i+1)));**
44. **}**
45. **printf("Function 9: e raised to n\n");**
46. **for(double i = 0;i<=100;i++) {**
47. **printf("%.3lf\n",exp(i));**
48. **}**
49. **printf("Function 10: 2 raised to log n\n");**
50. **for(double i = 0;i<=100;i++) {**
51. **printf("%.3lf\n",pow(2,log2(i)));**
52. **}**
53. **}**

**1.B-**

1. **#include <stdio.h>**
2. **#include<stdlib.h>**
3. **#include<time.h>**
4. **void main()**
5. **{**
6. **int n=0;**
7. **for(int k=0; k<(100000/100); k++)**
8. **{**
9. **n=n+100;**
10. **int num[n];**
11. **int insert[n];**
12. **int select[n];**
13. **int j, min;**
14. **clock\_t start\_t, end\_t;**
15. **double total\_t;**
16. **printf("%d\t",n);**
17. **for(int i=0; i<n; i++)**
18. **{**
19. **num[i]=rand() % 10;**
20. **insert[i]=num[i];**
21. **select[i]=num[i];**
22. **}**
23. **start\_t = clock();**
24. **for (int i = 1; i < n; i++)**
25. **{**
26. **int a = insert[i];**
27. **j = i - 1;**
28. **while (j >= 0 && insert[j] > a)**
29. **{**
30. **insert[j + 1] = insert[j];**
31. **j = j - 1;**
32. **}**
33. **insert[j + 1] = a;**
34. **}**
35. **end\_t = clock();**
36. **total\_t = (double)(end\_t - start\_t) / CLOCKS\_PER\_SEC;**
37. **printf("%f\t", total\_t );**
38. **start\_t = clock();**
39. **for (int i = 0; i < n; i++)**
40. **{**
41. **min = i;**
42. **for (j = i+1; j < n; j++)**
43. **{**
44. **if (select[j] < select[min])**
45. **{**
46. **min = j;**
47. **}**
48. **}**
49. **if(min != i)**
50. **{**
51. **int temp=select[i];**
52. **select[i]=select[min];**
53. **select[min]=temp;**
54. **}**
55. **}**
56. **end\_t = clock();**
57. **total\_t = (double)(end\_t - start\_t) / CLOCKS\_PER\_SEC;**
58. **printf("%f\n", total\_t );**
59. **}**
60. **}**

**Conclusion-**

**Thus I have understood the Insertion and Selection sort algorithm and their time complexities. I also understood how to calculate them and draw similar inferences.**