Name-Omkar deshmukh

div-CSE(DS)

UID-2021700018

Course-DAA

Exp-1B

Aim :- Experiment on finding the running time of an algorithm

Definition & Assumptions – For this experiment, you need to implement two sorting algorithms namely .Insertion and Selection sort methods. Compare these algorithms based on time and space complexity. Time required to sorting algorithms can be performed using high_resolution_clock::now() under namespace std::chrono. You have to generate1,00,000 integer numbers using C/C++ Rand function and save them in a text file. Both the sorting algorithms uses these 1,00,000 integer numbers as input as follows. Each sorting algorithm sorts a block of 100 integers numbers with array indexes numbers A[0..99], A[0..199], A[0..299],..., A[0..99999]. You need to use high_resolution_clock::now() function to find the time required for 100, 200, 300.... 100000 integer numbers. Finally, compare two algorithms namely Insertion and Selection by plotting the time required to sort 100000 integers using LibreOffice Calc/MS Excel. The x-axis of 2-D plot represents the block no. of 1000 blocks. The y-axis of 2-D plot represents the tunning time to sort 1000 blocks of 100,200,300,...,100000 integer numbers. Note – You have to use C/C++ file processing functions for reading and writing randomly generated 100000 integer numbers.

Sorting algorithm:

Sorting Algorithms Sorting Algorithms are a class of algorithms which are used to arrange the elements of a list or anarray in a particular order. There are a lot of sorting algorithms including bubble sort, selection sort, quick sort, insertion sort,heap sort, etc. This experiment focuses on insertion and selection sort.

our examples, we will consider sorting in ascending order.

Selection Sort:- The selection sort algorithm sorts an array by repeatedly finding the minimum element (consideringascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

1) The subarray which is already sorted.

2) Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element (considering ascending order) from theunsorted subarray is picked and moved to the sorted subarray.

Insertion Sort:- Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsortedpart are picked and placed at the

correct position in the sorted part. To sort an array of size n in ascending order: a. Iterate from arr[1] to arr[n] over the array. b. Compare the current element (key) to its predecessor. c. If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element

Program:-

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
//declaring function for insertion sort based on length
void insertion_sort(int arr[], int len){
int j = 1;
for (int i = 1; i < len; i += 1){
while ((arr[j] < arr[j-1]) && j!=0){
int temp = arr[j];
arr[j] = arr[j-1];
arr[j-1] = temp;
j -= 1;
}
j = i + 1;
}
}
//declaring function for selection sort based on length
void selection_sort(int arr[], int len){
int min = arr[0];
int m_ind = 0;
for (int i = 0; i < len; i +=1){
for (int j = i; j < len; j +=1){
if (arr[j] < min){
min = arr[j];
m_ind = j;
```

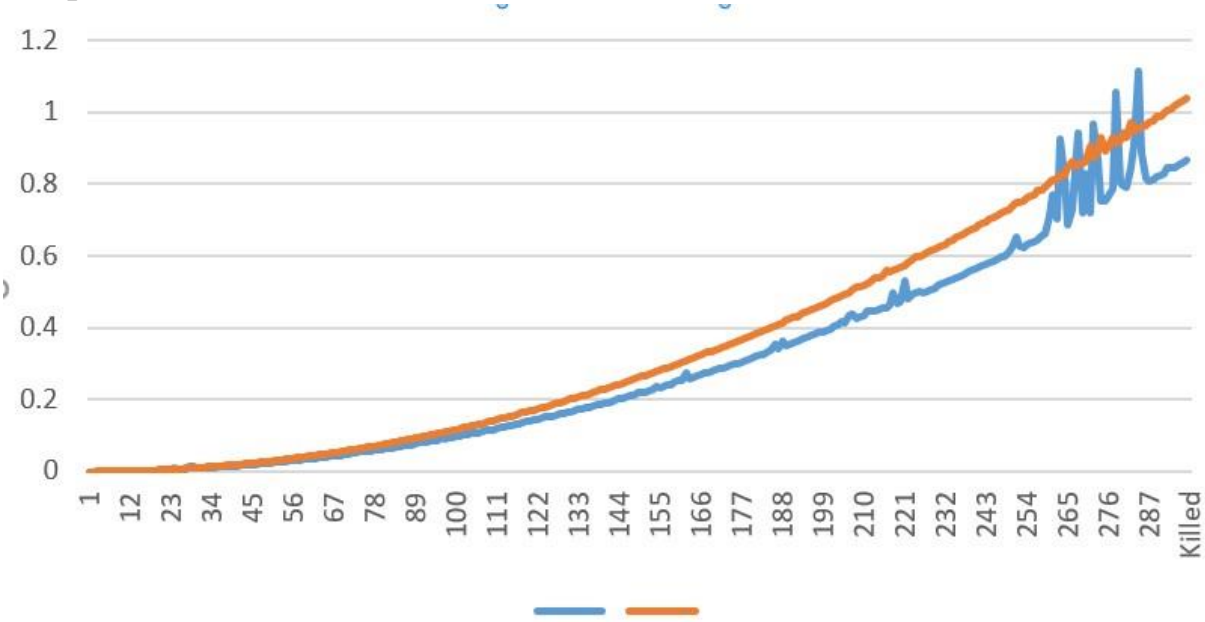
```
}
}
int temp = arr[m_ind];
arr[m_ind] = arr[i];
arr[i] = temp;
}
}
int main(){
//opening file to store input
FILE *fp = fopen("input.txt", "w");
if (fp == NULL)
{
printf("Error opening the file");
return -1;
}
//inputting 1 lakh random ints to input.txt
for (int i = 0; i < 100000; i += 1){
fprintf(fp, "%d ", rand());
}
fclose(fp);
//opening file to store output
FILE *fop = fopen("output.txt", "w");
if (fop == NULL)
{
printf("Error opening the file");
return -1;
}
//outputting code starts
int b = 1;
for (int j = 100; j < 100000; j += 100){
```

```
int arrs[j];
FILE *fir = fopen("input.txt", "r");
if (fir == NULL)
{
printf("Error opening the file");
return -1;
}
for (int k = 0; k < j; k +=1){
fscanf(fir, "%d ", &arrs[k]);
}
double t_selectsort = 0.0;
double t_insertsort = 0.0;
clock_t begin = clock();
selection_sort(arrs, j);
clock_t end = clock();
t_selectsort += (double)(end - begin) / CLOCKS_PER_SEC;
begin = clock();
insertion_sort(arrs, j);
end = clock();
t_insertsort += (double)(end - begin) / CLOCKS_PER_SEC;
fprintf(fop, "%d\t%f\t%f\n", b, t_insertsort, t_selectsort);
printf("%d\t%f\t%f\n", b, t_insertsort, t_selectsort);
b += 1;
fclose(fir);
}
fclose(fop);
}
```

Result:-

1	0.000024	0.000020
2	0.000084	0.000067
3	0.000199	0.000144
4	0.000374	0.000255
5	0.000556	0.000396
6	0.000809	0.000539
7	0.001092	0.000733
8	0.001456	0.000952
9	0.001395	0.001202
10	0.001750	0.001558
11	0.001535	0.001827
12	0.002847	0.002168
13	0.002170	0.002653
14	0.003139	0.002938
15	0.002819	0.003382
16	0.004063	0.003848
17	0.005483	0.004330
18	0.004696	0.004894
19	0.006510	0.005607
20	0.008747	0.006215
21	0.009174	0.006727
22	0.010479	0.007776
23	0.011757	0.008555
24	0.012359	0.008945
25	0.010802	0.009474
26	0.015117	0.010069
27	0.015370	0.010960
28	0.013762	0.012102
29	0.018443	0.013188
30	0.019838	0.013957
31	0.021702	0.015742
32	0.022041	0.015875
33	0.025876	0.016591
34	0.024624	0.019227
35	0.026623	0.018739
36	0.021942	0.020592
37	0.030663	0.021152
38	0.021107	0.023103
39	0.034396	0.022876

Graph:-



Observation: In this experiment, we wrote a program which given 100000 integers in 1000 blocks of 100 numbers, uses the selection andinsertion sort algorithms to sort them in ascending order. In selection sort, during every iteration, we move the minimum element of the unsorted subarray tothe end of the sorted subarray. In insertion sort, during every iteration, we move the first element of the unsorted subarray to the appropriate position in the sorted subarray.

Conclusion:-From this experiment I have learned about sorting. For implementing this program I have used selection sort and insertion sort. In thisb experiment I have generated 1,00,000 integer numbers using C/C++ Rand function and save them in a text file . And then by using these file ,each sorting algorithm sorts a block of 100 integers numbers with array indexes numbers.