**INTRODUCTION**

**NAME:** Nishkarsh Raj

**SAP ID:** 500060720

**REGISTRATION NUMBER:** R1712170-41

**BRANCH:** CSE DevOps (Xebia)

**BATCH:** B2

**SUBJECT:** Design And Analysis of Algorithm

**SUBJECT TEACHER:** Mr. Ahatsham

**EXPERIMENT 1**

**TITLE:** Brute Force Techniques

1. **Sort a given set of elements using bubble and selection sort and hence find the time required to sort elements**

**Code:**

#include<stdio.h>

#include<time.h>

void bubble\_sort(int A[],int n)

{

int i,j,temp;

for(i=0;i<n-1;i++)

{

for(j=0;j<n-i-1;j++)

{

if(A[j]>A[j+1])

{

temp = A[j+1];

A[j+1] = A[j];

A[j] = temp;

}

}

}

}

void selection\_sort(int A[],int n)

{

int i,j,temp;

for(i=0;i<n-1;i++)

{

for(j=i+1;j<n;j++)

{

if(A[j]<A[i])

{

temp = A[i];

A[i] = A[j];

A[j] = temp;

}

}

}

}

int main()

{

int n,i;

clock\_t st,en;

double t;

printf("Enter size of the array: ");

scanf("%d",&n);

int A[n];

for(i=0;i<n;i++)

{

printf("Enter the %d element of the array: ",i+1);

scanf("%d",&A[i]);

}

int B[n];

for(i=0;i<n;i++)

B[i]=A[i];

st = clock();

bubble\_sort(A,n);

en= clock();

t= (double)en-st/CLOCKS\_PER\_SEC;

printf("Time taken for Bubble sort is :%f \n",t);

st = clock();

selection\_sort(B,n);

en=clock();

t= (double)en-st/CLOCKS\_PER\_SEC;

printf("Time taken for Selection sort is :%f \n",t);

return 0;

}

1. **Perform linear search and find the time required to search an element.**

**Code:**

#include<stdio.h>

#include<time.h>

int linearsearch(int A[],int el,int n)

{

int i,flag=0;

for(i=0;i<n;i++)

{

if(A[i]==el)

{

flag=1;

break;

}

}

return flag;

}

int main()

{

int n,i,el,flag;

clock\_t start,end;

double time;

printf("Enter the number of elements in the array: ");

scanf("%d",&n);

int A[n];

for(i=0;i<n;i++)

{

printf("Enter the %d element of the array: ",i+1);

scanf("%d",&A[i]);

}

printf("Enter the element you want to search in the array: ");

scanf("%d",&el);

start=clock();

flag=linearsearch(A,el,n);

end=clock();

time = (double)start-end/CLOCKS\_PER\_SEC;

if(flag==1)

printf("The element exists in the array");

else

printf("The element does not exist in the array\n");

printf("Time taken for linear search is :%f\n",time);

return 0;

}

1. **Given a string called TEXT with ‘n’ characters and another string called PATTERN with ‘m’ characters (m<=n) .Write a program which implements brute force string matching to search for a given pattern in the text. If the pattern is present then find the position of first occurrences of Pattern in that Text.**

**Code:**

#include<stdio.h>

#include<time.h>

int search(char A[],char B[],int n,int m)

{

int i,j,pos=-1;

for(i=0;i<n-m;i++)

{

if(A[i]==B[0])

{

j=0;

while(j<m && A[i+j]==B[j])

j=j+1;

}

if(j==m)

pos=i;

}

return pos;

}

int main()

{

int n,m,pos;

clock\_t start,end;

double t;

printf("Enter the respective sizes of Text and Pattern arrays such that text has more size\n");

scanf("%d%d",&n,&m);

if (n>=m)

{

char TEXT[n],PATTERN[m];

printf("Enter the text string\n");

scanf("%s",TEXT);

printf("Enter the pattern string\n");

scanf("%s",PATTERN);

start=clock();

pos = search(TEXT,PATTERN,n,m);

end=clock();

t=(double)start-end/CLOCKS\_PER\_SEC;

printf("The element is found at %d index\n",pos);

printf("The time taken is %f\n",t);

}

else

{

printf("The size of pattern cannot be more than text!!");

}

return 0;

}