1.SELECTION SORT

CODE:

def selSort(arr,size):

for ind in range(size):

min=ind

for j in range(ind+1,size):

if arr[j]<arr[min]:

min=j

(arr[ind],arr[min])=(arr[min],arr[ind])

arr1=[-2, 45, 0, 11, -9,88,-97,-202,747]

size=len(arr1)

selSort(arr1,size)

print('The array after sorting in Ascending Order by selection sort is:')

print(arr1)

2. BUBBLE SORT

CODE:

def bubbleSort(arr):

n=len(arr)

for i in range(n-1):

swapped=False

for j in range(0,n-i-1):

if arr[j]>arr[j+1]:

swapped=True

arr[j],arr[j+1]=arr[j+1],arr[j]

if not swapped:

return

arr=[64,34,25,12,22,11,90]

bubbleSort(arr)

print("Sorted array is:")

for i in range(len(arr)):

print("%d"%arr[i],end=" ")

3.INSERTION SORT

def inSort(arr):

n=len(arr)

if n<=1:

return

for i in range(1,n):

key=arr[i]

j=i-1

while j>=0and key<arr[j]:

arr[j+1]=arr[j]

j-=1

arr[j+1]=key

arr=[12,11,13,5,6]

inSort(arr)

print(arr)

4.SEQUENTIAL SEARCH

CODE:

def search(arr,N,x):

for i in range(0,N):

if (arr[i]==x):

return i

return -1

if \_name=="main\_":

arr=[2,3,4,10,40]

x=10

N=len(arr)

result=search(arr,N,x)

if(result==-1):

print("Element is not present in array")

else:

print("Element is present at index", result)

5. STRING MATCHING USING BRITE FORCE

CODE:

def string\_search(string,pattern):

s\_len=len(string)

p\_len=len(pattern)

found=False

i=0

for i in range(s\_len-p\_len+1):

j=0

for j in range(p\_len):

if string[i+j]!=pattern[j]:

break

if j==p\_len-1:

found=True

break

if found:

print("Found pattern at index ", i)

else:

print("Could not find pattern")

def main():

string="ABCABAB ABABABAABAC"

pattern="ABABAABA"

string\_search(string, pattern)

if \_name=="main\_":

main()

6.CLOSET PAIR

CODE:

From math import sqrt

from random import randint

arr1=[]

dist=0

p1=[]

p2=[]

min1=1000

for i in range(0, 100):

arr1.append([randint(0,100),randint(0,100)])

print(arr1)

print("\n")

def dist(a,b):

x=pow((a[0]-b[0]),2)

y=pow((a[1]-b[1]),2)

return sqrt(x+y)

for i in range(0, len(arr1)):

for j in range(i+1, len(arr1)):

dis=dist(arr1[i],arr1[j])

if(dis<min1):

min1=dis

p1=arr1[i]

p2=arr1[j+1]

print(p1,"",p2,min1)

#print (sorted(arr1))

7.CONVEX HULL

CODE:

void convexHull(point \*array)

{

double a,b,c,checkVal,found;

for (int i = 0; i < 8; i++)

{

for (int j = i+1; j < 8; j++)

{

found = 0;

a = array[j].y - array[i].y;

b = array[j].x - array[i].x;

c = (array[i].x \* array[j].y) - (array[i].y \* array[j].x);

for (int k = 0; k < 8;k++)

{

checkVal = (a \* array[k].x) + (b \* array[k].y) - c;

if (checkVal == 0)

{

found = 1;

break;

}

}

if (found == 1)

{

printf("%lf %lf\n",nums[i].x,nums[j].y);

}

}

}

}

8.EXHAUSTIVE SEARCH

CODE:

def maxPackedSets(items, sets):

maxSets = 0

for set in sets:

numSets = 0

for item in items:

if item in set:

numSets += 1

items = [i for i in items if i != item]

maxSets = max(maxSets, numSets)

return maxSets

items = [1, 2, 3, 4, 5, 6]

sets = [

[1, 2, 3],

[4, 5],

[5, 6],

[1, 4]

]

maxSets = maxPackedSets(items, sets)

print(f"Maximum number of sets that can be packed: {maxSets}")