Lab Work

University of Baltistan Skardu

Submitted to:

Ma’am Noreen Maryam

Submitted By:

M Nawaz

Registration No:

S23BSCS014

Department:

Computer Science

Date:

March 27,2024

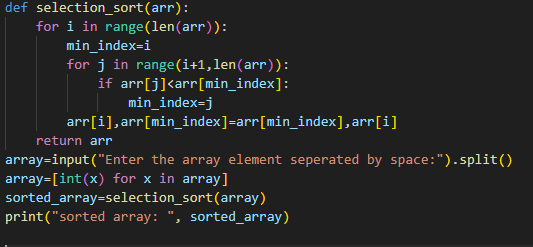
Section:

“A”

LAB#20

Q: Write a program that takes an unsorted array from the user and then sorts it by using selection sort. Explain each line of code in detail when you execute it.

Program:



Output:



Explanation:

Def- selection -sort ([3,6,1,8,2,4]):

n=len ([3,6,1,8,2,4])

for i in range(6): #i=0

minindex=0

for j in range(i+1,n):

if a[1]<a[0]:

6<3

X

for j in range(2,6):

if a[2]<a[0]

1<3

Minindex=2

for j in range(4,6):

if a[4]<a[2]

2<1

X

for j in range(5,6):

if a[5]<a[2]

4<1

X

a[0],a[2]=a[2],a[0]

a[0],a[2]=1,3

updated array:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 3 | 8 | 2 | 4 |

for i in range(6): #i=1

minindex=1

for j in range(1+1,n)

if a[2]<a[1]:

3<6

Minindex=2

for j in range(3,6):

if a[3]<a[2]

8<3

X

for j in range(4,6):

if a[4]<a[2]

2<3

True

Minindex=j

Minindex=4

for j in range(5,6):

if a[5]<a[4]

4<2

X

a[1],a[4]=a[4],a[1]

a[1],a[4]=2,6

updated array:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 8 | 6 | 4 |

for i in range(6): #i=3

minindex=3

for j in range(4,6):

if a[4]<a[3]:

6<8

minindex=4

for j in range(5,6):

if a[5]<a[4]:

4<6

Minindex=5

a[3],a[5]=a[5],a[3]

a[3],a[5]=4,8

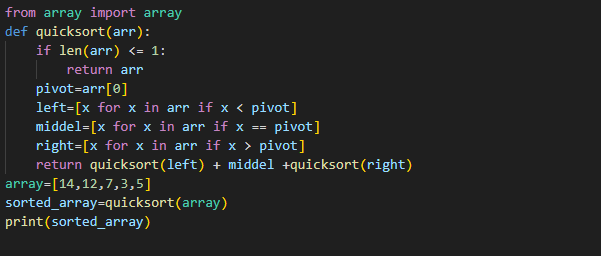
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 6 | 8 |

Thus the array is completely sorted in ascending order by the method of selection sort.

LAB#21

**Q.1:** Write a program to sort an unsorted array by using the quick sort algorithm consider first element as a pivot element.

Program:



Output:



Explanation:

**1. def quick\_sort(arr):**

**If len([14,12,7,3,5])<=1:**

**Return arr**

**Pivot=arr[0]**

**Pivot=14**

**Left= [12,7,3,5]**

**Middle= [14]**

**I return quick\_sort([12,7,3,5])+middle**

**1-left🡪quick\_sort([12,7,3,5])**

**Middle🡪14**

**quick\_sort([12,7,3,5])**

**1-left:**

**def quick\_sort(arr):**

**if len[12,7,3,5]<=1**

**X**

**Pivot=arr[0]**

**Left=7,3,5**

**Middle=12**

**Right=[]**

**II return quick\_sort[7,3,5]+middle+quick\_sort[]**

**1-left-left🡪quick\_sort[7,3,5]**

**1-left-middle🡪5**

**1-left-right🡪empty**

**1-left-left:**

**def quick\_sort(arr):**

**if len[7,3,5]<=1:**

**X**

**Pivot=arr[0]**

**Left=3,5**

**Middle=7**

**Right=[]**

**1-left-left-left=[3,5]**

**Midde=7**

**return quick\_sort[3,5]+middle+quick\_sort[]**

**1-left-left-left=[3,5]**

**def quick\_sort(arr):**

**if len[3,5]<=1:**

**X**

**Pivot=arr[0]**

**Left=[]**

**Middle=3**

**Right=[5]**

**1-left-left-left-right=5**

**return quick\_sort[]+middle+quick\_sort[5]**

**1-left-left-left-right=[5]**

**def quick\_sort(arr):**

**if len[5]<=1:**

**return 5**

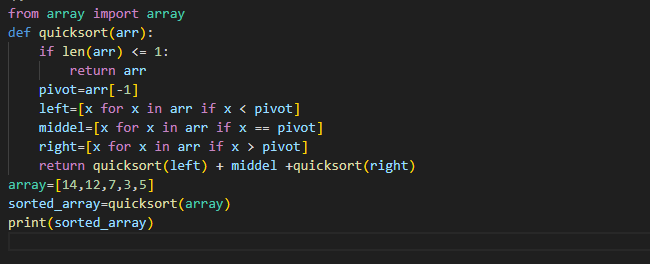
**I return quick\_sort([12,7,3,5])+14+quick\_sort()**

**return[3,5,7,12,14]**

**sorted\_a1=[3,5,7,12,14]**

**Q.2:** Write a program to sort an unsorted array by using the quick sort algorithm consider last element as a pivot element.

Program:



Output:



Explanation:

**1. def quick\_sort(arr):**

**If len([14,12,7,3,5])<=1:**

**Return arr**

**Pivot=arr[-1]**

**Pivot=5**

**Left=[3]**

**Middle=[5]**

**Right=[14,12,7]**

**I return quick\_sort([3])+middle+quick\_sort([14,12,7])**

**1-left🡪quick\_sort([3])**

**Middle🡪5**

**1-right🡪quick\_sort([14,12,7]**

**1-left:**

**def quick\_sort(arr):**

**if len[3]<=1:**

**return 3**

**Now:**

**I return quick\_sort[3]+middle+quick\_sort[]**

**quick\_sort([14,12,7])**

**1-right:**

**def quick\_sort(arr):**

**if len[14,12,7]<=1**

**X**

**Pivot=arr[-1]**

**Left=[]**

**Middle=7**

**Right=14,12**

**II return quick\_sort[]+7+quick\_sort[14,12]**

**1-right-left🡪empty**

**1-right-middle🡪7**

**1-right-right🡪14,12**

**1-right-right:**

**def quick\_sort(arr):**

**if len[14,12]<=1:**

**X**

**Pivot=arr[-1]**

**Left=[]**

**Middle=12**

**Right=14**

**1-right-right-right=14**

**1-right-right-middel=12**

**1-right-right-right:**

**def quick\_sort(arr):**

**if len[14]<=1:**

**return 14**

**now:**

**III return quick\_sort[]+7+quick\_sort[14,12]**

**Return [14,12]**

**We have:**

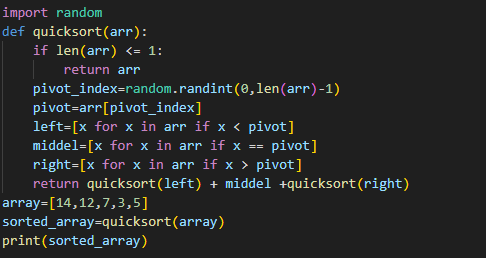
**1-right=[14,12]**

**I return quick\_sort([3])+middle+quick\_sort([7,12,14])**

**return[3,5,7,12,14]**

**sorted\_a1=[3,5,7,12,14]**

**Q.3:** Write a program to sort an unsorted array by using the quick sort algorithm consider any random element as a pivot element.



Output:



Explanation:

This program is written for random value of pivot. The value of pivot is randomly chosen but there is no difference on the output of the program.