

Bachenla Santhash

## LABORATORY WORK BOOK

demonde

TT P								Roll Number								
Course Code: AGS:D11 Course Name: D5 Laboratory							3	9	5 1	A	1	2	G.	3		
Nan	ne of the	Course Faculty Mo. K.	Laxmin	nariayai	mamma	2		V	Facul			RE	003	33		
Exe	rcise Nu	mber : 04	Week	Number :			11(9)	3		30	109	20	24			
S. No.	Exercise Number	EXERCISE NAME		Algorithm /	Source Code		T	Program Execution								
			Aim/ Preparation	Performanc	Calculations and Graphs			Results and Error Analysis			Viva - Voce	Total				
			4	-	4			4			4	2	0			
1	4.1	Puick Sort		3	K Sovt	ů.			eanls .			Aug)				
2	4.2	Meye Sort	t (inte	Will Soy	id a	V	3	1	西台	3	20	P				
3	4.3	Heap Sort	+01													
4	4.4	Radia Sort		Name :	(38)	A	5		000	1.						
5	4.5	Shell sort	(101 S	a olo			0		ø	,						
6		, Gard Tua	H	) padi	S	t	1		y			4	2	0		
7		2	(1 - %	Loud	(ann)	+1		K	divo							
8		. (3)	an ;	pî +1	HPO)	+	601	K	in the							
9							FÜ			t.	,					
10	- 6	ATTO THE POPULATION		1							1					
11		the associate house	1°) KO	tation	411	100	alo.	Hå	S)	ilo	19					
12		7 ( John +	0 48	1 × 110												

Santhon

Signature of the Student

· Avot = down [ law].

Signature of the Faculty

```
Divide And Conquer .
Quick Sort: -
                        people and alleghan
 AIM: - Write a Program to Dort the given
 list of elements using Juck Sort.
PROGRAM: -
Public class quick Sort &
                                     Evol Hind Lit
  Public Static void quick Sort (int[] our, Int low,
                             int high) {
      if (low < high) {
        int pi = Partition (aver, low, high).
        quick Sort ( over, low, pi-1);
        Quick Sort ( av, pi+1, high);
   3
   Public Static int partition (intil au, int low,
                              int high ) {
      int Pivot = ovn [low];
```

```
for (int j = low + 1; j <= high; j++) {
  if (our [i] < pivot) {
  Dwap (ave, i, j);
                 14 1 = an ! [ength ]
     1++:
   System. out. printer ("Original away: "):
Dwap (av, low, i-1):
return :-1; (1-1,0,000) +100 xbilly
     System out , printly (" Sorted avery")
Public Static void Dwap (int[] avec, inti, inti) {
  int temp = aur [i];
  aux[i] = aux[j].
 wor [j] = temp.
TYPUT: ann = (7,6,10,5,9,2,15,7]
Public State void print Array (ant is own) {
 for (int i=0; i < avr. length; i++) {
 System. out: print ( avr [i] + " ");
```

```
System. out. println();
    for (int i - low + 1; iz= ligh; i++) {
 Public Static void main (String [] args) !
   int[] over = {7,6,10,5,9,2,1,15,73;
   int n = our length;
   System. out. printin ("Original away:");
   Point Array (avu);
                 . (1 -1 wal , row) gows
   vuich Sort (ave, 0, n-1);
   System. out. println ("Sorted averay?")
Print Array (avi)
                     int temp = apriling.
                      oser[i] = oser[i]:
RESULT: -
                      000([i] = temp;
INPUT: ann = [7, 6, 10, 5, 9, 2, 15, 7]
OUTPUT: au = [2,5,6,7,7,9,10,15]
       for (int i=0; i < and . Jameth; i++) {
           elystem, out : print ( wor [:] + " ").
```

```
(((()) <= R(()) (
Merge Sort :-
AIM: - Write a Program to Dort the given
list of elements using Merge Sort.
PROGRAM: -
                       and [K] = KIII.
class Merge Sort (
 void mange (int avoil7, int d, int m, intr)
 { int \eta_1 = m-1+1;
  int 72 = Y-m;
  int LCI = new int[n1];
                          ann [K] = [ [1] ;
  int RCJ = new int [12].
  for (int = 0; i < n1; +++)
    [[i] = au [] +i].
  for (int j=0; j<n2; ++j) } (sn> b) alife
   R[j] = avu [m+l+j]; [i] A = [A]
  int i = 0, j = 0;
  int K = 1;
  While (i < n1 && j < n2) {
```

1.4

```
if (Lci) <= Rci) {
      out [K] = L[i]:
AIM: - White a program to doct the given
     list of elements using Menge Sort.
    else 1
                              PROGRAM: -
     OUT [K] = R[j]:
                           ) type spro Menals
     j++ :
   int mange (int anni), int b, int m int ) }
               me q1 = m-1+1; 1 (1-1)
   K++ .
                     int 12 = 1-m;
  While (i < 11) {
                  : [[1] +02 (10) = [] +ni
   WH [K] = [[i];
                   igt 860 = new intens);
  i++;
               for ( int 1= 0; izn1; +++)
   K++;
                    ·[i+2] rua =[i].
While (1 < 12) { (1++ : 21 > 1 : 0 - 1 + 11) rot
OUN [K] = R[j] ; [,+ l+m] was = [i,] i
  j++ :
                      int i= 0, i=0;
  K++ :
                         int Keli
            VIII (1201 && 3202) +
4
```

```
void sort (int world, int l, int r)
    Marge Sort ob = gen Margy So(vsl) fi
       int m = (1+1)/2.
       Dort (au, l, m);
      Dort (au, m+1, r); and yours true?
       menge ( ave, l, m, r);
                                  A ESULT : -
  Dtatic void print Array (int avoic)
  int \eta = aver. Jangth; ii + 2 = 1000; rugtio
    for (int i=0; i<n; ++i) : +vol qual;
     Egotem. out. print ( ovi [i] + " ");
System. out. println();
        list of elements using Heap Sort.
  Public Dtatic void main (String arys [7)
   int our [] = { 12, 11, 13, 5, 6, 7 3;
   System. out. println (" Given Array");
```

4.3

```
Pount Array (aux); +11 12 1000 +11 1000 +11
     Merge Sort ob = new Merge Sort ();
     Ob. Dort (over, 0, avr. Jength - 1);
     System. out. println ("In Sorted avery").
    Pount Array (avr).
                 mouse ( one, l, m, Y).
RESULT : -
INPUT: OUL = [12, 11, 13, 5, 6, 7]
OUTPUT: DUL = [5, 6, 7, 11, 12, 13]
Heap Sort: - (itt insi one this) not
         Extem. out. print ( autij + " ")
AIM: - Write a Program to Sout the given
list of elements wring Heap Sort.
PROGRAM: - side side side side side
Public class Heapsort 1 1 22 1 1 1 1000
   Public void Sort (int aurcs)
```

8/16

ROLL NUMBER:

```
int N = our. length;
 for (int i=N|2-l; i=0; i--)
   heapify (av, N, i)
 for (int i= N-l; 170; 1-7) f
   int temp = over[0];
   hispity (and M. Small : [i] was = [0] was
   our [i] = temp;
  heapify (ave, i, 0);
                : Herol - rose = 11 Jni }
3
void heapify (int over [], int N, inti)
int largest = i; mi) thing, the metals
 int 1 = 2 * i + 1; () patrice , two morals
 int Y = 2 * i + 2;
 if (l<N && ar [l] > arr [largest])
  largest = 1; [1 82] = [31000 +11
if (Y<N & & aur [r] > our [largest])
```

```
largest = Y; Atomil non = 11 this
 if (largest ! = i) ( ... - 2 Mai + ii) 10;
  int Dwap = over [i];
  aun [i] = ann [langest];
  ann [langest] = Dwap:
  heapity (ou, N, largest);
Dtate void print Array (int our [])
int N = avr. length;
  for (intij=0; 12 N; ++i)
   elystem. out. print (aus [i] +"11).
 System. out. println ().
Public Statie void main (Stung augs [])
 int our [] = {12, 11, 13, 5, 6, 7 }
int N = our longth;
```

```
HenpSort ob = new HeapSort();
      Ob. Sout (ann);
      System. out. println (" Sorted arouge is");
      Print Array (ove);
                      · [i] race = sm
  3
 RESULT :-
 OUTPUT: Sorted away is 5,6,7,11,12, 13.
 Rodiz Sort: - (1) this was - (1) through this
 AIM: - Write a program to Don't the given
  list of elements using Radix Sort.
 PROGRAM:-
 import. java. 10. *; (qx3 /[i] row) ] troop
 import java. util. *.
 class Radix (:[1-1] trous = + [i] trous
                  for ( != n - 1 : ! >= 0 :
Dtate int get Max (int over [7, int n)
```

```
theresort ob a view floor; [0] now = am this
  for (int i=1; i<0n; i++)
  if (aucij > mx)
                    Print Array ( wow);
   mx = wor [i];
 Veturn mx:
Static void count Sort ("int own[], int n, int exp.)
int output [] = new int[n];
                         +A Hodix Sort :-
 int count[] = new int[10];
 Arrays . fill (count, 0).
for (i = 0; i < n; i++)
  count [ (avn [i] / exp) % 10) ++;
for (i= 1; i < 10; i++) +. lite. ouoi troqui
  count [i] + = count [i-1]; allow and
for (i=n-1; i=0; i--)
 output [count[(ovn(i]/exp) % 10] - 1] = ovn(i);
```

```
count [ (aus [i] / exp) % 10] --;
   for (i=0; i<n; i++)
     our [i] = output [i];
  Static void radia Sort (int avoil), int n)
     int m = get Max (ave, n);
                              : type lole
     for (int exp = 1; m | exp > 0; exp* = 10)
=ountSoyt (aus, n, exp);
       tras 110 8 primer Demonds 30 tall
  State void print (int aux[], intn); MARION
    for (int i=0; i<n; i++) = 0.00.00.00.00
     System. out. print (avoi [i] + "")
  3
 Public Static void main (Strung [] args)
   int over[] = ( 170, 45, 75, 90, 802, 24, 2, 66 };
   int in = over. length;
   Vaolizsort (au, n);
```

```
Prients (aux, n); al as (and little and )
                   (++1:0 = 1 · 0 = 1) 10}
RESULT : -
                  · [i] tustus = [i] .
INPUT: WOW = [ 170, 45, 75, 90, 802, 24, 2, 66]
OUTPUT: OUT = [2,24,45,66,75,90,170,802]
                nt m = get Max (ans, n):
Shell Sort :-
AIM: - Write a Program to Dort the given
list of elements using Shell Sort.
PROGRAM: - (1) to to to the state of
 =lass ShellSort
  Static void print Array (int over [7)
  int n = aver. length; som box state sider.
   for (int i=0; i<n; ++i)
 System. out. print (over [i] +" ");
  System. out. println();
                        (11 100) tracklost
```

```
int Sout (Int works)
                     * (374) good 2 163
  int n = au. langth;
  for (int gap = 1/2; gap >0; gap | = 2)
  for (int i= gap; i < n; i += 1)
     int temp = avefij;
      int j;
      for (j=i; j > = gap && aun [j-gap] >
          temp; j = gap)
                               JOSO : TONTO
        arr[j] = aur [j - gap];
       our [i] = temp.
  Yetwon 0.
j
Public Statie void main (String args[])
int wur[] = {12,34,54,2,33;
```

```
System. out. println (" Array before Douting");
  Print Array (aux);
                     int ne and Jonath .
  Shell Sort ob = new Shell Sort();
  Ob . Dort (wer);
  System. out. printin (" Array after Dorting");
  Pount Array (avr);
                 :[i] web = qmst fri
 3
RESULT: -
INPUT: OU = [12, 34, 54, 2, 3]
OUTPUT: aux = [2,3,12,34,54]
                 · dwap = 1; X row
                              et muter
   (Clapes private) mion bior stold silder
      int aux (12, 34, 54, 2, 33;
```

VIVA VOCE :-

A)

A)

- 1) What is Sorting Algorithm?
  - A Sorting Algorithm is used to reasonage a given away (or) list elements wing to Compare Operator on the elements. The Comparison Operator is used to decide the new Order of element in the respective Data Structure.
  - 2) Explain about Merge Sort?
    - Merge Sort is a general purpose Sorting technique purely based on Divide & Conquer Approach. In the Divide & Conquer Technique, the etements are divided into smaller parts of lists. Then the appropriate function is applied to each half of the main input list. Further, the halves are Merged together to get the Yesult.
    - How does puick Sort Work?

3)

puick Sort is Divide & Conquer Sorting Algorithm. It chooses a pivot element & rearrange the Away so that elements are smaller than the Pivot are on the left and elements greater are on the right.

What is Radix Sort?

Radix Sort is the Sorting Technique which is based on the Value. In this method, the elements can be sorted by making use of either of the two methods LSD (Least Significant Digit)

MSD (Most Significant Digit).

the the stant of

Xan