**KIET Group of Institutions, Ghaziabad**

Department of Computer Applications

(An ISO – 9001:2015 Certified & ‘A’ Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

**Experiment-no**

**Objective:** Implement the Heap Sort algorithm to sort the given list of N numbers and plot graph

|  |  |  |
| --- | --- | --- |
| Scheduled Date | Compiled Date | Submitted Date |
| 25-09-20 | 25-09-20 | 25-09-20 |

Algorithm of Heap Sort

Heapsort(A)

1.BUILD-MAX-HEAP

2. for 🡨 length[A] downto 2

3. do exchange A[I] <--> A[j]

4. heap-size[A] 🡨 heap-size[A]- 1

5. MAX-HEAPIFY(A,I)

Program of Heap Sort

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

int count=0;

void max\_heapify(int \*,int);

void build\_max\_heap(int \*,int);

void heapsort(int \*,int);

void swap(int \*,int \*);

int heapsize;

void main()

{

void getdata(int[],int);

void putdata(int[],int);

int \*arr,n,i;

clrscr();

printf("enter size of array = ");

scanf("%d",&n);

arr=(int \*)malloc(sizeof(int)\*n);

getdata(arr,n);

printf("\n Unsorted array =");

putdata(arr,n);

heapsort(arr,n);

printf("\n sorted array=");

putdata(arr,n);

printf("\n count=%d", count);

getch();

}

0 void heapsort(int \*arr,int len)

{

int i;

count++;

build\_max\_heap(arr,len);

count++;

for(i=len-1;i>=1;i--)

{

count++;

swap(&arr[0],&arr[i]);

count++;

heapsize = heapsize -1;

count++;

max\_heapify(arr,0);

count++;

}

}

void max\_heapify(int \*arr,int i)

{

int l=2\*i,r=2\*i+1,largest;

count++;

if(l<heapsize && arr[l]>arr[i])

{

count++;

largest = l;

count++;

}

else

{ count++;

largest = i;

count++;

}

if(r<heapsize && arr[r]>arr[largest])

{

count++;

largest = r;

count++;

}

count++;

if (largest != i)

{

count++;

swap(&arr[i],&arr[largest]);

count++;

max\_heapify(arr,largest);

count++;

}

}

void build\_max\_heap(int \*arr,int len)

{

heapsize = len;

int i;

count++;

for(i = len/2;i>=0;i--)

{

count++;

max\_heapify(arr,i);

count++;

}

}

void swap(int \*a,int \*b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void getdata(int x[20], int n)

{

int k;

printf("\n enter the %d elements for sorting=\n",n);

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

{

scanf("%d",&x[k]);

}

}

void putdata(int x[20],int n)

{

int k;

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

{

printf("%d",x[k]);

}

printf("\n");

}

Heap sort graph

|  |  |  |  |
| --- | --- | --- | --- |
| Input size | Best case | Average Case | Worst Case |
| 5 | 133 | 133 | 87 |
| 10 | 348 | 330 | 275 |
| 15 | 593 | 488 | 450 |
| 20 | 870 | 828 | 780 |