**Ex. No. 8 Binary Heap**

**Date:**

**Aim:**

To build a binary heap from an array of input elements.

**Algorithm:**

1. Start

2. In a heap, for every node x with parent p, the key in p is smaller than or equal to the key in x.

3. For insertion operation

a. Add the element to the bottom level of the heap.

b. Compare the added element with its parent; if they are in the correct order, stop.

c. If not, swap the element with its parent and return to the previous step

4. For deleteMin operation

a. Replace the root of the heap with the last element on the last level.

b. Compare the new root with its children; if they are in the correct order, stop.

c. If not, Swap with its smaller child in a min-heap

5. Stop

**Program:**

/\* Binary Heap \*/

#include <stdio.h>

#include <limits.h>

int heap[1000000], heapSize;

void Init()

{

heapSize = 0;

heap[0] = -INT\_MAX;

}

void Insert(int element)

{

heapSize++;

heap[heapSize] = element;

int now = heapSize;

while (heap[now / 2] > element)

{

heap[now] = heap[now / 2];

now /= 2;

}

heap[now] = element;

}

int DeleteMin()

{

int minElement, lastElement, child, now;

minElement = heap[1];

lastElement = heap[heapSize--];

for (now = 1; now \* 2 <= heapSize; now = child)

{

child = now \* 2;

if (child != heapSize && heap[child + 1] < heap[child])

child++;

if (lastElement > heap[child])

heap[now] = heap[child];

else

break;

}

heap[now] = lastElement;

return minElement;

}

void main()

{

int number\_of\_elements;

printf("Program to demonstrate Heap:\nEnter the number of number\_of\_elements:");

scanf("%d", &number\_of\_elements);

int iter, element;

Init();

printf("Enter the elements: ");

for (iter = 0; iter < number\_of\_elements; iter++)

{

scanf("%d", &element);

Insert(element);

}

for (iter = 0; iter < number\_of\_elements; iter++)

printf("%d ", DeleteMin());

printf("\n");

}

**Result:**

Thus a binary heap is constructed for the given elements