DATA STRUCTURES

DAY-10

1. c program for binary heap

Program

```
#include <stdio.h>
#define MAX_SIZE 100
int heap[MAX_SIZE];
int size = 0;
void swap(int *a, int *b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}
void heapify(int i) {
   int largest = i;
   int left = 2 * i + 1;
   int right = 2 * i + 2;
   if (left < size && heap[left] > heap[largest])
      largest = left;
   if (right < size && heap[right] > heap[largest])
      largest = right;
   if (largest != i) {
      swap(&heap[i], &heap[largest]);
      heapify(largest);
   }
```

```
}

void insert(int value) {
    if (size == MAX_SIZE) {
        printf("Heap is full. Cannot insert more elements.\n");
        return;
    }
    size++;
    int i = size - 1;
    heap[i] = value;
    while (i != 0 && heap[(i - 1) / 2] < heap[i]) {
        swap(&heap[i], &heap[(i - 1) / 2]);
        i = (i - 1) / 2;
    }
}
int extractMax() {
    if (size <= 0)
        return -1;
    if (size == 1) {
        size--;
        return heap[0];
    }
    int root = heap[0];
    heap[0] = heap[size - 1];
    size--;
    heapify(0);
```

```
    return root;
}

int main() {
    insert(3);
    insert(2);
    insert(15);
    insert(5);
    insert(4);
    insert(45);
    printf("Max element extracted: %d\n", extractMax());
    return 0;
}
```

Output:

```
Max element extracted: 45
```

2.Heap Sort

Program:

```
#include <stdio.h>
#include <stdlib.h>
 void swap(int *x, int *y) {
    int temp = *x;
    *x = *y;
    *y = temp;
}
```

```
 void heapify(int arr[], int n, int i) {
   int largest = i;     // Initialize largest as root
   int left = 2 * i + 1; // left = 2*i + 1
   int right = 2 * i + 2; // right = 2*i + 2
    if (left < n && arr[left] > arr[largest])
       largest = left;


      if (right < n && arr[right] > arr[largest])
       largest = right;
   if (largest != i) {
       swap(&arr[i], &arr[largest]);
       heapify(arr, n, largest);
   }
}
 void heapSort(int arr[], int n) {
   // Build heap (rearrange array)
   for (int i = n / 2 - 1; i >= 0; i--)
       heapify(arr, n, i);
    for (int i = n - 1; i > 0; i--) {
            swap(&arr[0], &arr[i]);
        heapify(arr, i, 0);
   }
}
 void printArray(int arr[], int n) {
   for (int i = 0; i < n; i++)
       printf("%d ", arr[i]); printf("\n");
```

```
}
int main() {
   int arr[] = {12, 11, 13, 5, 6, 7};
   int n = sizeof(arr) / sizeof(arr[0]);
   printf("Original array: \n");
   printArray(arr, n);
   heapSort(arr, n);

   printf("Sorted array: \n");
   printArray(arr, n);
   return 0;
}
```

Output:

Original array:

12 11 13 5 6 7

Sorted array:

5 6 7 11 12 13