

ATULYA RAI

BT22CSH009

ASSIGNMENT - 5

1)

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  void heapify(vector<int> &arr, int n, int i)
5  {
6      int smallest = i;
7      int left = 2 * i + 1;
8      int right = 2 * i + 2;
9      if (left < n && arr[left] < arr[smallest])
10         smallest = left;
11      if (right < n && arr[right] < arr[smallest])
12         smallest = right;
13      if (smallest != i)
14      {
15         swap(arr[i], arr[smallest]);
16         heapify(arr, n, smallest);
17      }
18  }
19  void buildHeap(vector<int> &arr)
20  {
21      int n = arr.size();
22      for (int i = n / 2 - 1; i >= 0; i--)
23      {
24         heapify(arr, n, i);
25      }
26  }
27  void insert(vector<int> &arr, int value)
28  {
29      arr.push_back(value);
30      int index = arr.size() - 1;
31      while (index > 0 && arr[index] < arr[(index - 1) / 2])
32      {
33         swap(arr[index], arr[(index - 1) / 2]);
34         index = (index - 1) / 2;
35      }
36  }
37  int main()
38  {
39      vector<int> arr = {1, 5, 6, 8, 9, 7, 3};
40      buildHeap(arr);
```

```

41     cout << "Min Heap: ";
42     for (int num : arr)
43     {
44         cout << num << " ";
45     }
46     cout << endl;
47     int newValue = 4;
48     insert(arr, newValue);
49     cout << "Min Heap after inserting " << newValue << ": ";
50     for (int num : arr)
51     {
52         cout << num << " ";
53     }
54     cout << endl;
55     return 0;
56 }

```

OUTPUT

```

PS C:\programs> cd "c:\programs\" ; if ($?) { g++ comp.cpp -o comp } ; if ($?) { .\comp }
Min Heap: 1 5 3 8 9 7 6
Min Heap after inserting 4: 1 4 3 5 9 7 6 8

```

2)

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  void heapify(vector<int> &arr, int n, int i, bool isMinLevel)
5  {
6      int largest = i;
7      int left = 2 * i + 1;
8      int right = 2 * i + 2;
9      if (isMinLevel)
10     {
11         if (left < n && arr[left] < arr[largest])
12             largest = left;
13         if (right < n && arr[right] < arr[largest])
14             largest = right;
15     }
16     else
17     {
18         if (left < n && arr[left] > arr[largest])
19             largest = left;
20         if (right < n && arr[right] > arr[largest])

```

```

21         largest = right;
22     }
23     if (largest != i)
24     {
25         swap(arr[i], arr[largest]);
26         heapify(arr, n, largest, !isMinLevel);
27     }
28 }
29 int deleteMax(vector<int> &arr)
30 {
31     if (arr.empty())
32     {
33         cerr << "Heap is empty!" << endl;
34         return -1; // Return a sentinel value to indicate an empty heap.
35     }
36     int maxElement = arr[0];
37     int lastIndex = arr.size() - 1;
38     swap(arr[0], arr[lastIndex]);
39     arr.pop_back();
40     bool isMinLevel = true; // Root is at the min-level
41     heapify(arr, arr.size(), 0, isMinLevel);
42     return maxElement;
43 }
44 int main()
45 {
46     vector<int> minMaxHeap = {9, 8, 6, 7, 5, 1, 3};
47     cout << "Max Element Deleted: " << deleteMax(minMaxHeap) << endl;
48     cout << "Remaining Min-Max Heap: ";
49     for (int num : minMaxHeap)
50     {
51         cout << num << " ";
52     }
53     cout << endl;
54     return 0;
55 }
56

```

OUTPUT

```

PS C:\programs> cd "c:\programs\" ; if ($?) { g++ comp.cpp -o comp } ; if ($?) { .\comp }
Max Element Deleted: 9
Remaining Min-Max Heap: 3 8 6 7 5 1
PS C:\programs>

```

3)

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  void heapify(vector<int> &arr, int n, int i)
5  {
6      int largest = i;
7      int left = 2 * i + 1;
8      int right = 2 * i + 2;
9      if (left < n && arr[left] > arr[largest])
10         largest = left;
11     if (right < n && arr[right] > arr[largest])
12         largest = right;
13     if (largest != i)
14     {
15         swap(arr[i], arr[largest]);
16         heapify(arr, n, largest);
17     }
18 }
19 void buildMaxHeap(vector<int> &arr)
20 {

```

```

21     int n = arr.size();
22     for (int i = n / 2 - 1; i >= 0; i--)
23     {
24         heapify(arr, n, i);
25     }
26 }
27 void heapSort(vector<int> &arr)
28 {
29     int n = arr.size();
30     buildMaxHeap(arr);
31     for (int i = n - 1; i > 0; i--)
32     {
33         swap(arr[0], arr[i]);
34         heapify(arr, i, 0);
35     }
36 }
37 int main()
38 {
39     vector<int> arr = {12, 11, 13, 5, 6, 7};
40     cout << "Original Array: ";

```

```

41     for (int num : arr)
42     {
43         cout << num << " ";
44     }
45     cout << endl;
46     heapSort(arr);
47     cout << "Sorted Array: ";
48     for (int num : arr)
49     {
50         cout << num << " ";
51     }
52     cout << endl;
53     return 0;
54 }

```

OUTPUT

```
PS C:\programs> cd "c:\programs\" ; if ($?) { g++ comp.cpp -o comp } ; if ($?) { .\comp }  
Original Array: 12 11 13 5 6 7  
Sorted Array: 5 6 7 11 12 13  
PS C:\programs>
```