**­Q1**

**template<class T>**

**void BinaryHeap<T>::ReheapUp(int root, int last)**

**{**

**int parent;**

**if (last > root) // tree is not empty**

**{**

**parent = (last - 1) / 2;**

**if (data[parent] < data[last])**

**{**

**Swap(data, parent, last);**

**ReheapUp(root, parent);**

**}**

**}**

**}**

**template<class T>**

**void BinaryHeap<T>::ReheapDown(int cnode, int last)**

**{**

**int maxChild, rightChild, leftChild;**

**leftChild = 2 \* cnode + 1;**

**rightChild = 2 \* cnode + 2;**

**if (leftChild <= last) // left child is part of the heap**

**{**

**if (leftChild == last) // only one child**

**maxChild = leftChild;**

**else**

**{**

**if (data[leftChild] <= data[rightChild])**

**maxChild = rightChild;**

**else**

**maxChild = leftChild;**

**}**

**if (data[cnode] < data[maxChild])**

**{**

**Swap(data, cnode, maxChild);**

**ReheapDown(maxChild, last);**

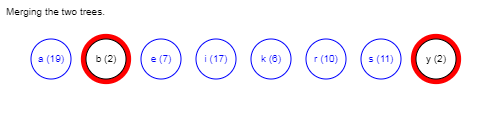
**}**

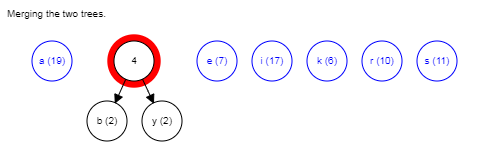
**}**

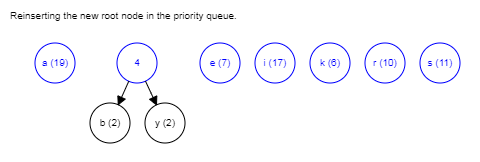
**}**

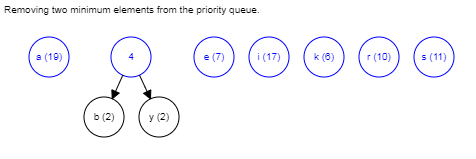
**Q2**

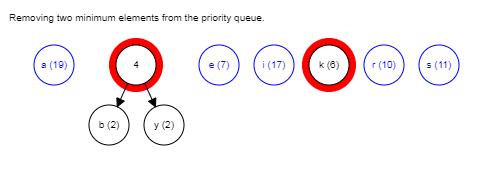


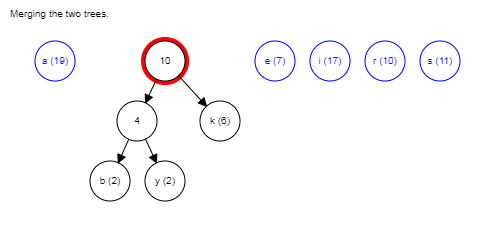


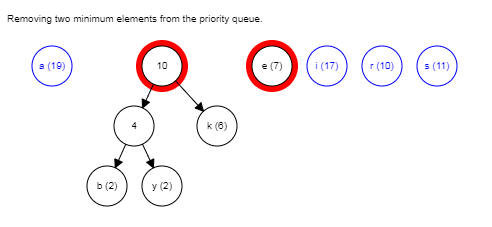


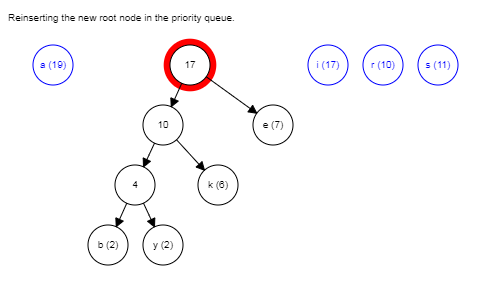


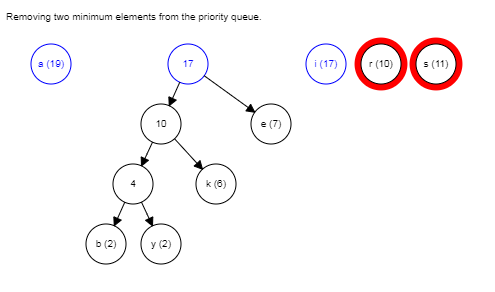


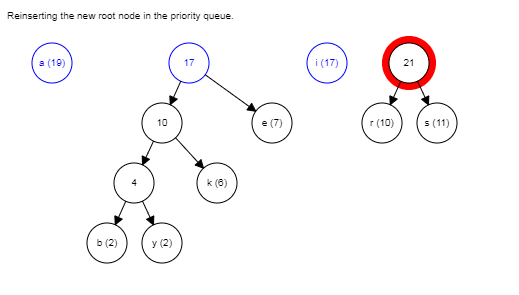


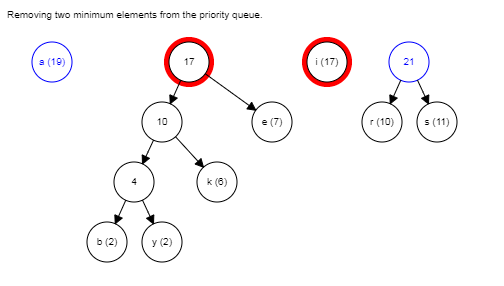


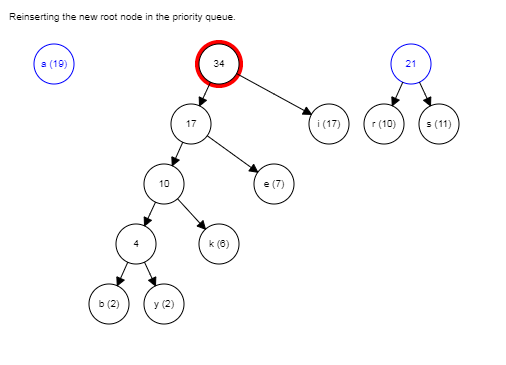


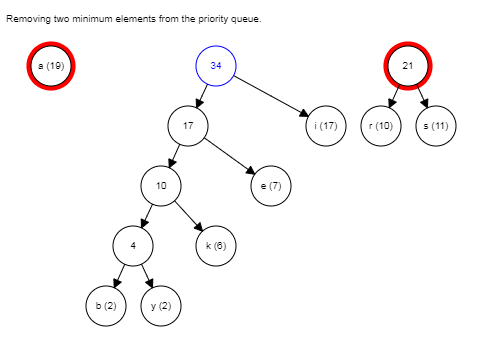


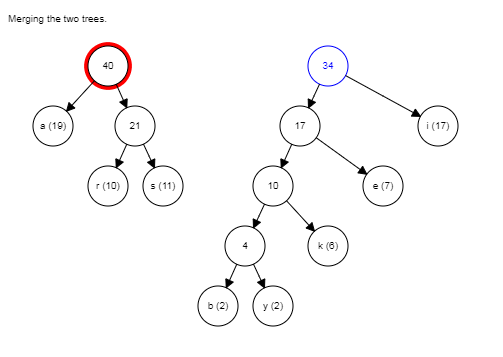


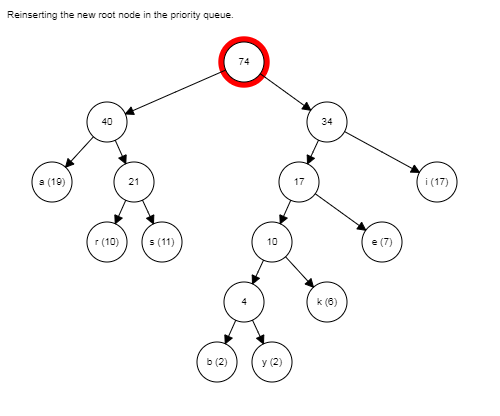












**Each step -----------------------5\*(1) marks**