***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

***BATCH : B11***

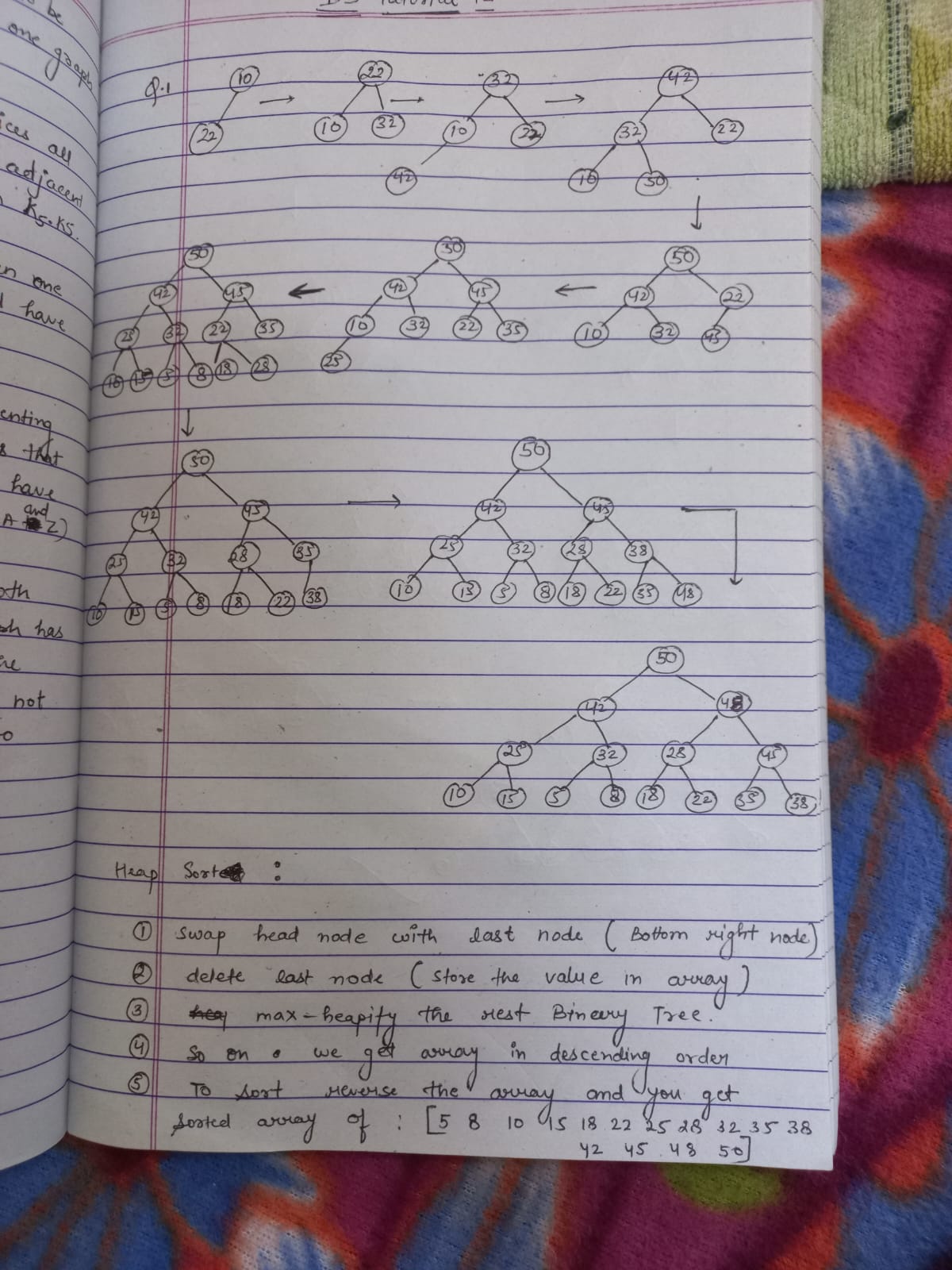
***Data Structure [15B11CI311]***

***Tutorial Sheet***

***Week 12***

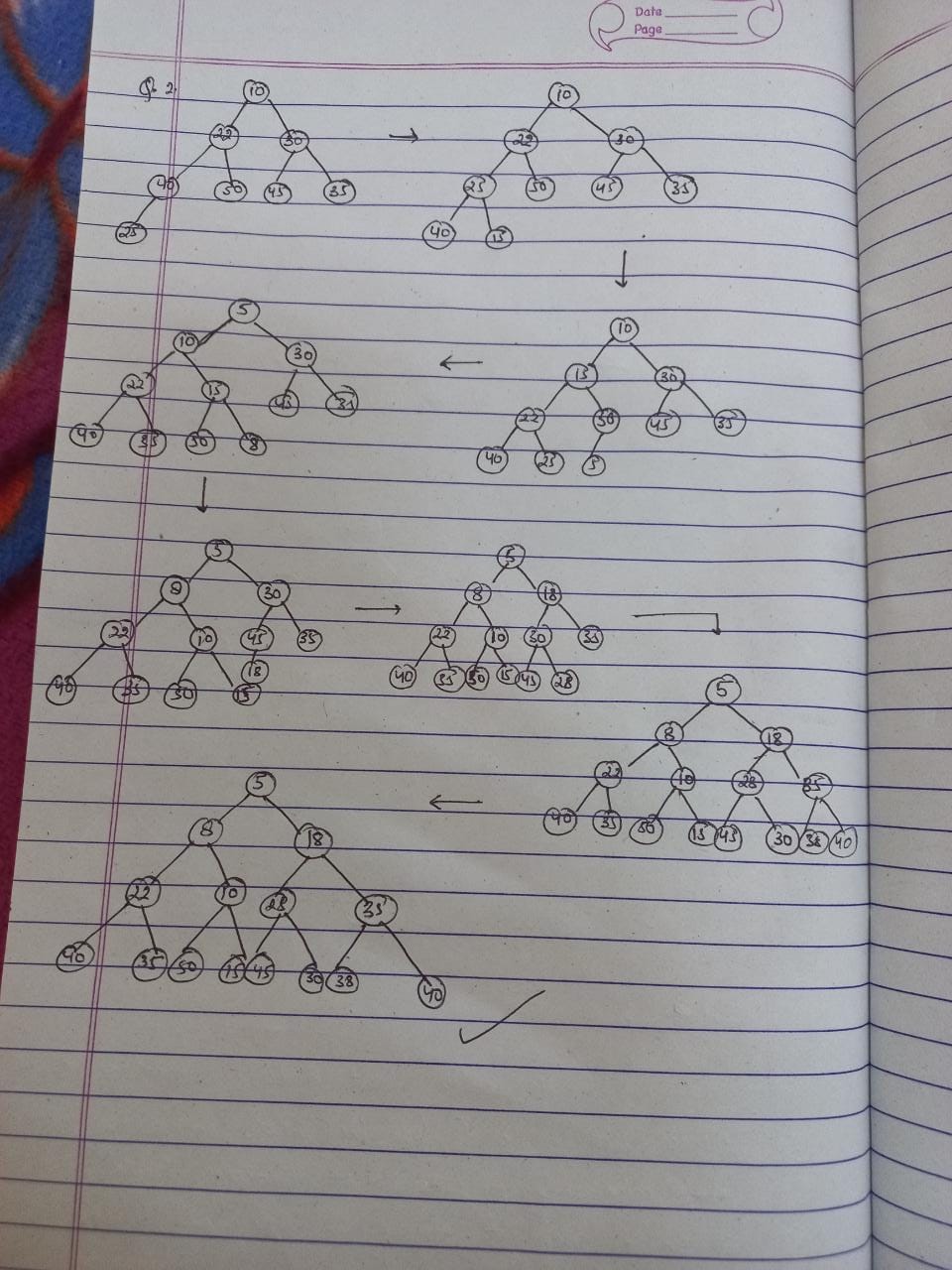
*Q.1 Construct a Max Binary Heap using following elements: 10, 22, 32, 42, 50, 45, 35, 25, 15, 5, 8, 18, 28, 38, and 48 and sort them in ascending order using heap sort.*

***Solution :-***

**

*Q2. Construct a Min Binary Heap using following elements: 10, 22, 30, 40, 50, 45, 35, 25, 15, 5, 8, 18, 28, 38, and 48*

***Solution :-***

**

*Q3. Analyse the performance of the Binary Heap, if it is created using (a) array and (b) binary tree (complete)*

***Solution :-***

*Binary heap creating with array is better than creating with binary tree. Because it takes less space as binary tree gas pointers which takes more space.*

*Q4. You have been given a Max Binary Heap and it is desired to convert it into a Min Binary Heap. Write a program to perform the desired task.*

***Solution :-***

*#include <bits/stdc++.h>*

*using namespace std;*

*void MinHeapify(int arr[], int i, int N)*

*{*

*int l = 2 \* i + 1;*

*int r = 2 \* i + 2;*

*int smallest = i;*

*if (l < N && arr[l] < arr[i])*

*smallest = l;*

*if (r < N && arr[r] < arr[smallest])*

*smallest = r;*

*if (smallest != i)*

*{*

*swap(arr[i], arr[smallest]);*

*MinHeapify(arr, smallest, N);*

*}*

*}*

*void convertMinHeap(int arr[], int N)*

*{*

*for (int i = (N - 2) / 2; i >= 0; --i)*

*MinHeapify(arr, i, N);*

*}*

*void printArray(int \*arr, int size)*

*{*

*for (int i = 0; i < size; ++i)*

*cout << arr[i] << " ";*

*}*

*int main()*

*{*

*int arr[] = {20, 18, 10, 12, 9, 9, 3, 5, 6, 8};*

*int N = sizeof(arr) / sizeof(arr[0]);*

*printf("Max Heap array : ");*

*printArray(arr, N);*

*convertMinHeap(arr, N);*

*printf("\nMin Heap array : ");*

*printArray(arr, N);*

*return 0;*

*}*

*Q5. You have been given two Max Binary Heaps and it is desired to merge both heaps into a single heap. Write a program to perform the desired task.*

***Solution :-***

*#include <bits/stdc++.h>*

*using namespace std;*

*void maxHeapify(int arr[], int N, int idx)*

*{*

*if (idx >= N)*

*return;*

*int l = 2 \* idx + 1;*

*int r = 2 \* idx + 2;*

*int max = idx;*

*if (l < N && arr[l] > arr[idx])*

*max = l;*

*if (r < N && arr[r] > arr[max])*

*max = r;*

*if (max != idx)*

*{*

*swap(arr[max], arr[idx]);*

*maxHeapify(arr, N, max);*

*}*

*}*

*void buildMaxHeap(int arr[], int N)*

*{*

*for (int i = N / 2 - 1; i >= 0; i--)*

*maxHeapify(arr, N, i);*

*}*

*void mergeHeaps(int merged[], int a[], int b[], int N,*

*int M)*

*{*

*for (int i = 0; i < N; i++)*

*merged[i] = a[i];*

*for (int i = 0; i < M; i++)*

*merged[N + i] = b[i];*

*buildMaxHeap(merged, N + M);*

*}*

*int main()*

*{*

*int a[] = {10, 5, 6, 2};*

*int b[] = {12, 7, 9};*

*int N = sizeof(a) / sizeof(a[0]);*

*int M = sizeof(b) / sizeof(b[0]);*

*int merged[N + M];*

*mergeHeaps(merged, a, b, N, M);*

*for (int i = 0; i < N + M; i++)*

*cout << merged[i] << " ";*

*return 0;}*