Mir hossain

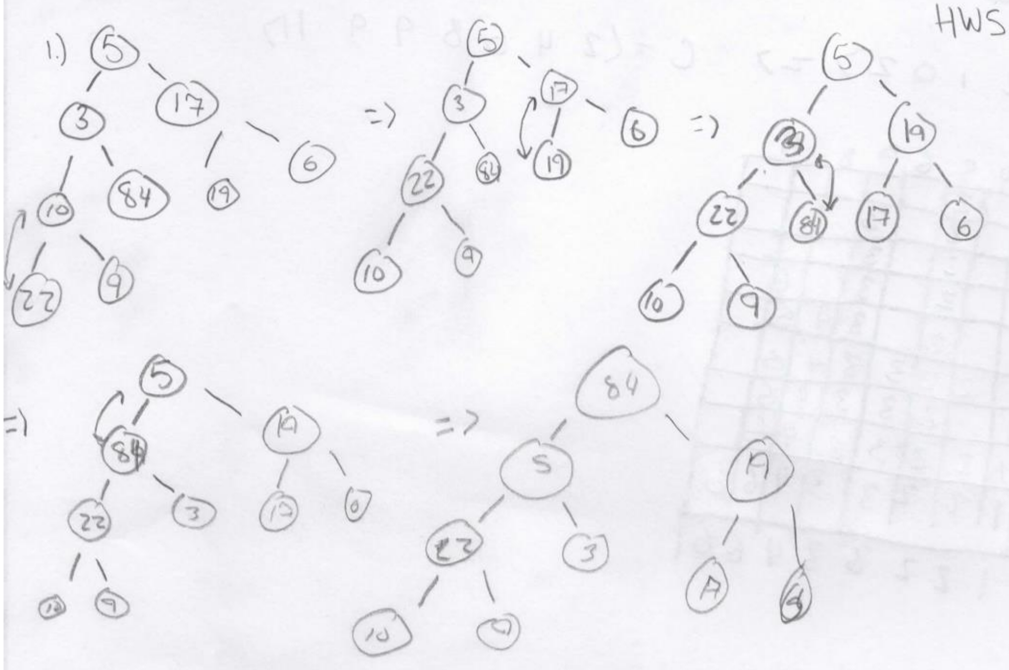
Cs477

Hw5

1A.) 6.3-1

Miv Hossain
CS477
HW5

Create Max Heap on <5, 3, 17, 10, 84, 19, 6, 22, 9>



2.) 8.2-3

The algorithm would be correct regardless of the order. However, the algoritm isn't steble. In the final for loop, a repeating element that apperers later in the array is placed in the output array. In the original algorithm, a repeating element that appecrs earier, would be placed before an repeating element that appears later in the array.

↳ A[1, 5, 5, 6] => B => First 5 should be placed first.

↑ this is 8.2-3, 1B →

↓ Note to grader* 8.2-1 will be on the next page

1B.) Illustrate Counting Sort

A = 6 0 2 0 1 3 4 6 1 3 2

C = <2 2 2 2 1 0 2> => C = <2 4 6 8 9 9 11>

B

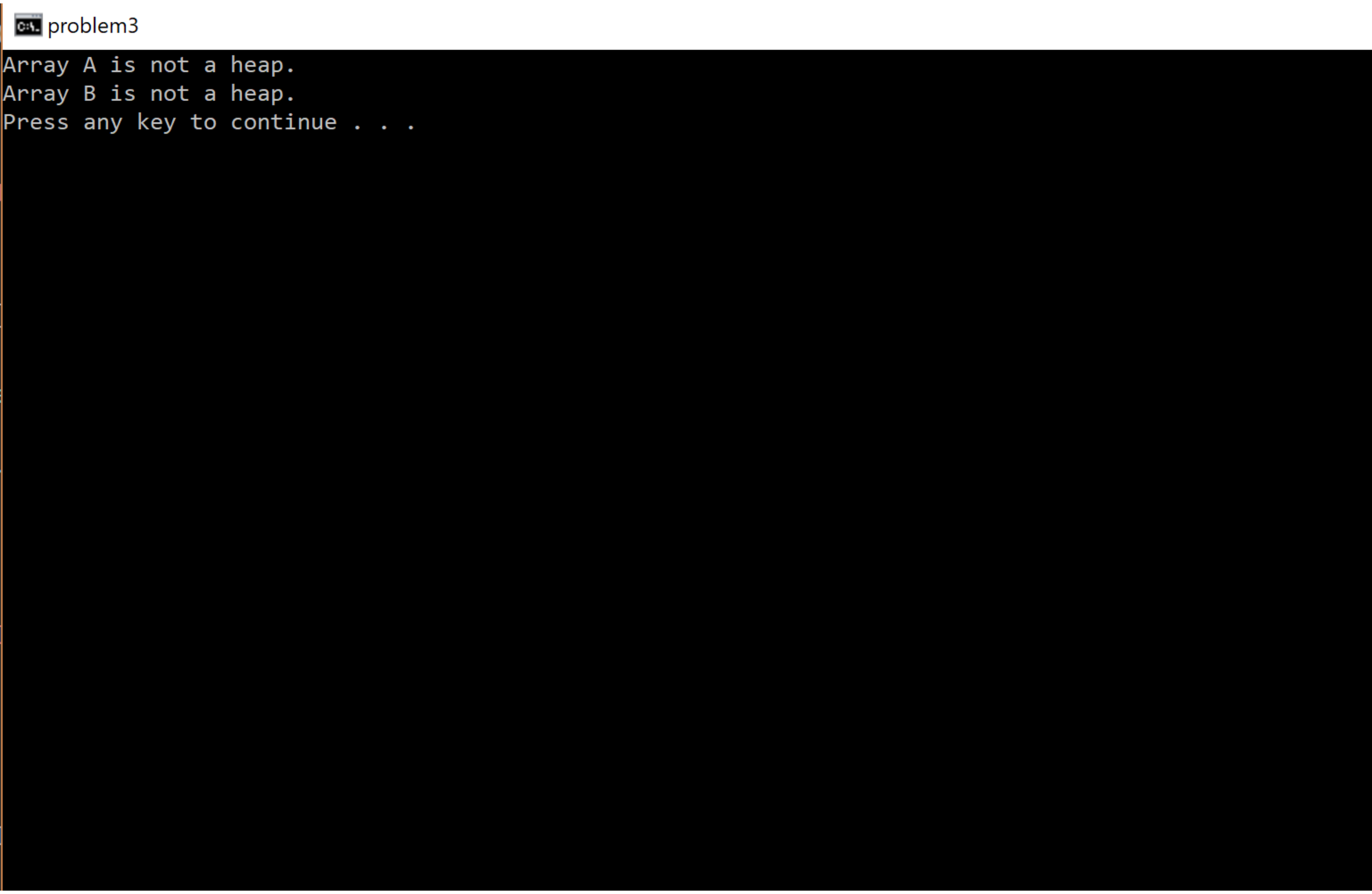
0	1	2	3	4	5	6	7	8	9	
					2					
					2		3			
					2		3			
			1		2		3			6
			1		2		3	4		6
			1		2	3	3	4		6
			1		2	3	3	4		6
		1	1		2	3	3	4		6
	0	1	1	2	2	3	3	4	6	6
0	0	1	1	2	2	3	3	4	6	6

6 0 1 1 2 2 3 3 4 6 6

3.) On seperate File

4.) 6.3-2

We want the loop in index i to decrease from ⌊A.length/2⌋ to 1 because every node would be satisfied. If we increased from 1, then the root node would be called first and not retraced. Every node would be satisfied except the root node if we increased by 1 at A.length/2.



```
Array A is not a heap.
Array B is not a heap.
Press any key to continue . . .
```

5.) 8.2-4 (197)

Describe an algorithm that gives n ints in range 0-k, preprocesses its input, then answers any query about how many of the n ints fall into the range [a...b] in O(1) Algorithm should use Θ(n+k) preprocessing time

Given : Array A[1...n], range is 0-k

1.) Two new arrays
↳ B[0..K];
C[0...k];

2.) Every element in B, set to 0
B = [0..k] = {0};

3.) Make for loop to increment C respective to elements in A array

```
↳ 3A.) for (i = 1 -> A.length()   / O(n) complexity
   3B.) B[A[i]) = B(A[i]) + 1
   3C.) C[0] = B[0]
```

4.) use for loop to store count to Jth element in B[ ] array

```
4A.) for (j = 1 -> k)   / O(k) time
4B.) C[j] = B(j] + C[j-1]
```

↳ Complexity is ⊢ O(n+k); C[b] - C[a] + B[a] => O(1) after pre-processing

3