

Solutions to Problem 1 of Homework 4 (10 points)

Name: GOWTHAM GOLI (N17656180)

Due: Tuesday, October 6

Using a min-heap in a clever way, give $O(n \log k)$ -time algorithm to merge k sorted arrays $A_1 \dots A_k$ of size n/k each into one sorted array B . Write the pseudocode of your algorithm using procedures BUILD-HEAP, EXTRACT-MIN and INSERT.

Solution:

Algorithm

- Build a Heap using the first elements of the given arrays A_1, \dots, A_k
- Extract Minimum from the heap and insert into B
- Pick the next element from the array, the popped element of the heap came from and insert into this heap
- Keep repeating until all the elements of the given arrays are inserted into the heap

Pseudocode

```
1 Algorithm: MERGEHEAPS( $A_1, \dots, A_k, B$ )
2  $Heap \leftarrow$  BUILD-HEAP( $A_1[0], \dots, A_k[0]$ )
3  $B \leftarrow$  NEWARRAY( $n$ )
4  $c \leftarrow 0$ 
5 while  $Heap$  is not empty do
6    $A_i[p] \leftarrow$  EXTRACT-MIN( $Heap$ )
7    $B[c] \leftarrow A_i[p]$ 
8   if  $p < n/k$  then
9     INSERT( $A[p+1], Heap$ )
10  end
11   $c++$ 
12 end
13 Return  $B$ 
```

Algorithm 1: Merge sorted arrays A_1, \dots, A_k using a min-heap

Time Complexity

In the first step, time taken to build heap of size k is $O(k)$. Time taken to extract minimum from the heap and then insert into the heap is $O(\log k)$ and this is done n times, therefore $O(n \log k)$. Therefore, total time takes is $O(k) + O(n \log k) = O(n \log k)$

□