CS3230 Design and Analysis of Algorithms

Homework 1
TA: Mohammed Haroon Dupty (haroon@u.nus.edu)

15-10-2018

Instructions:

- Due by: 11:59 pm 28-10-2018 (Sunday)
- Solutions must be in a single pdf file and uploaded to the workbin in IVLE.
- Name the file in this format : Studentid_Name.pdf e.g A0000000A_Ali.pdf
- Please write the following on the first page:
  - Full name
  - Source, if you obtained the solution through some external source
- You are allowed to discuss, but must write up your own assignment.

1 K-Sorted Array

We say that an array of distinct integers $A[1..n]$ is k-sorted if for each $1 \leq i \leq n$, the i'th smallest element is in positions $A[1], ..., A[i+k]$. Given below is the pseudocode for sorting a k-sorted array using heap as an auxilary data structure.

```
Algorithm 1: Sort K-sorted array
Input: A $k$-sorted array $A[1,\ldots,n], k$
Output: Sorted array of A
Initialize array B[1,...,n]
S = MakeHeap($A[1],\ldots,A[k]$)
for  $j = k+1$ to $n+k$ do
    if $j \leq n$ then
        Insert(S, $A[j]$)
    B[$j-k$] ← ExtractMin(S)

return $B$
```

Algorithm 1 uses two auxilary data structures, array B and heap S. It puts $k+1$ elements in heap and extracts minimum from the heap iteratively while adding a new element from array A into the heap. We want to prove correctness of the algorithm from this intuition: *given that first t elements in array B are in their "correct" place, then $t+1^{th}$ element must also be "correct".*

Work through the following questions to show that this algorithm correctly sorts the array.

1.1

Make a statement, capturing the above intuition, on the value extracted from heap at line 6.

1.2

Prove the statement you made in 1.1

1.3

Now using induction on elements of array B, prove the correctness of the algorithm.

2 Inversions

Let $A[1 \dots n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A.

2.1

List the five inversions of the array $\langle 2, 3, 8, 6, 1 \rangle$

2.2

What array with elements from the set {1, 2, . . . , n} has the most inversions? How many does it have?

2.3

Give an algorithm that determines the number of inversions in any permutation on n elements in $\Theta(n \log n)$ worst-case time. (Hint: Modify merge sort.)