Exercise Sheet 6
COMS10017 Algorithms 2020/2021

Reminder: $\log n$ denotes the binary logarithm, i.e., $\log n = \log_2 n$.

1 Big-O Notation

Rank the following functions by order of growth: (no proof needed)

$$(\sqrt{2})^{\log n}, n^2, n!, (\log n)!, (\frac{3}{2})^n, n^3, \log^2 n, \log(n!), 2^{2^n}, n \log n$$

Hint: Stirling's approximation for the factorial function can be helpful:

$$e(\frac{n}{e})^n \leq n! \leq en(\frac{n}{e})^n$$

2 Decision Trees

1. Give a lower bound on the number of queries needed for sorting 4 elements.
2. Give an optimal decision tree/guessing strategy for sorting 4 elements a, b, c, d (draw the decision tree).
3. How many comparisons does the Insertionsort algorithm make in the worst case when sorting an array of length 4?

3 kth Smallest Element

Give an algorithm that runs in time $O(n + k \log n)$ that computes the kth largest number in an array of n distinct integers.

Hint: Think about Heapsort!

4 Sorting

We are given an array A with $n + m$ elements so that the first n elements are sorted and the last m elements are unsorted.

1. What is the runtime of Insertionsort on array A?
2. Suppose that $m = O(1)$. How can we sort A as efficiently as possible and what is the resulting runtime?
3. Suppose that $m = O(\sqrt{n})$. How can we sort A as efficiently as possible and what is the resulting runtime?

4. What is the largest value of m so that we can obtain a runtime of $O(n)$?

5. Suppose that $m = \Theta(n)$. How can we sort A as efficiently as possible and what is the resulting runtime?

5 Optional and Difficult Questions

Exercises in this section are intentionally more difficult and are there to challenge yourself.

5.1 A Different Type of Sorting Algorithm

Consider the following algorithm for sorting an array A of size n:

1. Sort recursively the first $2/3$ of A, i.e., $A[0, \ldots, 2/3n - 1]$
2. Sort recursively the last $2/3$ of A, i.e., $A[n/3 - 1, n - 1]$
3. Sort recursively the first $2/3$ of A, i.e., $A[0, \ldots, 2/3n - 1]$

Answer the following questions:

1. Argue/prove that the algorithm really sorts A.
2. What is the runtime of A?