Algorithmic Combinatorics
Tibor Szabó
Anurag Bishnoi

Wi 2018–19

Exercise Sheet 1

Due date: 16:15, 23rd October

You should try to solve all of the exercises below, but clearly mark which two solutions you would like us to grade – each problem is worth 10 points. We encourage you to submit in pairs, but please remember to indicate the author of each solution. Until submission, you are forbidden to look at the solutions of any of these exercises on the internet.

Exercise 1 Consider the following algorithm to find the minimum of a set of $n = 2^k$ distinct numbers.

Algorithm: MIN
Input: $A = \{a_1, \dots, a_n\}$, $a_i \neq a_j \; \forall i \neq j$, $n = 2^k \geq 2$
Result: $\text{MIN}(A) = \min\{a_1, \dots, a_n\}$

```
if n = 2 then
    if a_1 < a_2 then
        return a_1;
    else
        return a_2;
    end
else
    for 1 ≤ i ≤ n/2 do
        set y_i = MIN({a_{2i-1}, a_{2i}});
    end
    return MIN({y_1, ..., y_{n/2}});
end
```

(a) Show that the MIN algorithm requires $n-1$ comparisons to find the minimum element, and that this is the best possible.

(b) After running the MIN algorithm to find the minimal element, how many additional comparisons are required to find the second-smallest element?

(c) After running the MIN algorithm to find the minimal element, how many additional comparisons are required to find both the second and the third smallest elements?

Exercise 2 There is a machine that takes as input at most 5 integers and outputs these numbers in sorted order. If using this machine once counts as a step, then determine the minimum number of steps you need to find the

(a) largest number from a set of 25 numbers;

(b) largest three numbers from a set of 25 numbers.

Exercise 3 Consider the following game. I think of an integer x between 1 and n, and your job is to try and determine x. You are allowed to ask questions of the form "Is $x < a$?" or "Is $x > a$?" for any a in $[n]$.

(a) Show that you can find x with only $\lceil \log_2 n \rceil$ questions, and that this is best possible.

To make your job slightly harder, I am now allowed to lie to you at most k times, for some constant k.

(b) How many questions do you now need to determine x? Provide the best lower and upper bounds that you can find.

Exercise 4 Give an algorithm with worst case time-complexity $O(n \log n)$ that given a set S of n real numbers, and a real number a, determines whether there exist two elements of S that add up to a.

Exercise 5 A DAG is a directed graph G which has no directed cycles, i.e., no (non-empty) directed path that starts and ends at the same vertex. In a depth/breadth first search tree, a back-arc is an edge (u, v) of G such that v is an ancestor of u in the tree.

(a) Prove that a directed graph is a DAG if and only if there exists a depth first search tree of this graph which contains no back arc.

(b) Prove that the previous statement is not true if you look at the breadth first search trees.

Exercise 6 Given a connected graph G and an arbitrary vertex $v_0 \in V(G)$, show that the breadth-first search starting at v_0 returns a tree T_B that preserves distances[1] to v_0; that is, for every $v \in V(G)$, $d_G(v_0, v) = d_{T_B}(v_0, v)$.

[1] In an unweighted graph $G = (V, E)$, the *distance* $d_G(u, v)$ between vertices $u, v \in V$ is the minimum length of a path from u to v. If u and v are in separate connected components, we may take their distance to be infinite.