1. Werkcollege 1

You are allowed to answer in Dutch.

1.1. Suppose we have a computer which can perform 1 million ($= 10^6$) operations per second. The seven formulas below denote the running time of some algorithms (measured in number of operations) depending on the number of elements n we feed to the algorithm. Determine for each algorithm how many elements can be processed in 1 minute.

a) n^2 *b*) $n \log n$ *c*) 2^n *d*) $n\sqrt{n}$ *e*) n^{100} *f*) 4^n *g*) n

1.2. Are the following statements true or not? No proof is required.

a) $n + 1 \in \mathcal{O}(n)$
b) $2n \in \mathcal{O}(n)$
c) $n + \log n \in \mathcal{O}(n)$
d) $n\sqrt{n} \in \mathcal{O}(n \log n)$
e) $n^2 + 2n + 1 \in \mathcal{O}(n^2)$
f) $2^{n+1} \in \mathcal{O}(2^n)$
g) $2^{2n} \in \mathcal{O}(2^n)$
h) $3^n \in \mathcal{O}(n2^n)$

1.3. Recall that in order to prove $f \in \mathcal{O}(g)$ one has to choose $c > 0$ and n_0 and then prove that $f(n) \leq cg(n)$ for all $n \geq n_0$.

a) Prove (2.a) and (2.b).
b) Prove that $n \in \mathcal{O}(2^n)$. (Hint: use $c = 1$ and $n_0 = 1$, then prove the inequality with induction.)

1.4. Consider the following algorithm.

```
search(int v[], int n, int x) {
    int i = 0
    bool found = false
    while (i < n && !found) {
        if (v[i] == x) {
            found = true
        }
        i++
    }
    if (found) return i-1;
    else return -1;
}
```

a) What is the worst case scenario? How many operations does it take in this case (your answer should depend on n)?

b) What is the best case scenario? How many operations does it take in this case (your answer should depend on n)?

1.5. For each item, give the least $\mathcal{O}$-class it belongs to. Justify your answer.

a) $T(n) \leq T(n-1) + 3n$
b) $T(n) \leq T(\frac{n}{2}) + 42$
c) $T(n) \leq 3T(\frac{n}{3}) + 13$
d) $T(n) \leq 4T(\frac{n}{4}) + n$

1.6. You are given two sorted arrays of size m and n. Give an $\mathcal{O}(\log m + \log n)$ time algorithm for computing the kth smallest element in the union of the two arrays.

1.7. Given a sorted array of distinct integers $A[1 \cdots n]$, you want to find out whether there is an index i for which $A[i] = i$. Give a divide-and-conquer algorithm that runs in time $\mathcal{O}(\log n)$.

Some facts (no exercises)

Some tips and tricks (most of these are not needed for this particular exercise sheet, but might come in handy at some point during the course). Summations:

$$1+2+\cdots+n = \sum_{i=0}^{n} i = \frac{n(n+1)}{2} \qquad 1+r+\cdots+r^n = \sum_{i=0}^{n} r^i = \frac{r^{n+1}-1}{r-1}$$

If these are not familiar to you, try to prove them yourself (for example with induction), practising such proofs is always useful ;-). About logarithms and exponentials:

$$2^0 = 1 \qquad 2^1 = 2 \qquad 2^a 2^b = 2^{a+b} \qquad 2^{ab} = (2^a)^b = (2^b)^a$$
$$\log_2(1) = 0 \qquad \log_2(2) = 1 \qquad \log_2(ab) = \log_2(a) + \log_2(b)$$

If the logarithm rules are unfamiliar to you, try proving them from the exponentiation rules and the identity:

$$\log_2(2^x) = x \qquad \text{for all } x \in \mathbb{R}.$$