

## Solution to Homework 33

**Problem.** Suppose that we have a probabilistic algorithm that gives a correct answer  $3/4$  of the time. How many times do we need to repeat this algorithm to make sure that the probability of a false answer does not exceed 3%? Give an example of a probabilistic algorithm. Why do we need probabilistic algorithms in the first place?

**Solution.** According to the lecture, if after one iteration, the probability of error is  $p_0$ , then after  $k$  iterations, the probability of error is  $p_0^k$ . In our case,  $p_0 = 1 - \frac{3}{4} = \frac{1}{4}$ , so the probability of an error after  $k$  iterations is  $\left(\frac{1}{4}\right)^k = \frac{1}{4^k}$ .

We want to find the smallest value  $k$  for which  $\frac{1}{4^k} \leq 3\% = \frac{3}{100}$ . The function  $1/x$  is decreasing for  $x > 0$ , thus the desired inequality is equivalent to  $4^k \geq \frac{100}{3} = 33.33\dots$

We all know the powers of 2 and thus, powers of 4:  $4^2 = 16$  is smaller than  $33.33\dots$ , but  $4^3 = 64$  is already larger than  $33.33\dots$ . So, the answer is that we need to repeat this algorithm  $k = 3$  times.

**Example** of a probabilistic algorithm can be taken from the lecture: to check whether two functions  $f(x)$  and  $g(x)$  are identical, we compare the values  $f(r_i)$  and  $g(r_i)$  of these two functions at one or more random points  $r_1, \dots, r_k$ .

- If at least for one of the random numbers  $r_i$ , we get  $f(r_i) \neq g(r_i)$ , we conclude that the two given functions are different.
- If  $f(r_i) = g(r_i)$  for all  $i = 1, \dots, k$ , then we conclude that the functions  $f(x)$  and  $g(x)$  are most probably identical – but we understand that they may still be different.

**Why do we need probabilistic algorithms?** Many problems are NP-complete, meaning that no feasible algorithm is possible that solves all the instances of such a problem. Since we cannot have a solution that works always, a natural next idea is to have a solution that works with some high probability.