# Algorithm analysis

1. Describe the RAM model. What assumptions are made and how well does it match an actual modern computer?

The RAM model is an approximation of a modern computer (which is roughly a universal Turing machine) with unbounded memory, although it violates some of the constraints of a “normal” computer. It, for example, doesn’t take into account that multiplication is a set of additions, that accessing memory takes different amounts of time depending on the type of memory etc.

1. Name and describe two other important models of computation.

Two other important types of models of computation are the Turing machine and the counter machine.

1. What does the Big-Oh notation mean? You should both have an intuitive understanding for what O(f(n)) means and know what it means to a mathematician.
2. In the following, what statements are true?
   1. 2n + 17 is in O(n). True
   2. 10n is in O(n^2). True
   3. n^2 + log n is in O(n^2) True
3. Why do we talk to “time complexity” but report the (asymptotic) number of steps of an algorithm?

We want to insure ourselves against many different kinds of input sizes.

1. Consider the problem of multiplying two n-digit integers. What is the time complexity this problem, if we take single-digit multiplication and addition as the elementary operations?

The multiplication of two n-digit numbers (a and b) will yield a polynomial on the form of (a\_1 + a\_2 + … + a\_n)(b\_1 + b\_2 + … b\_n) with the decimal positions of the two numbers as separate terms in each factor. First a\_1 will have to be multiplied with each of the terms in the factor containing b. Then each subsequent term in a will be multiplied this way as well. Then each term in the factor containing b will be multiplied to each term in the factor containing a. For each increment in n this will result in 2n^2+n-1.

https://en.wikipedia.org/wiki/Karatsuba\_algorithm