Algorithms Worksheet 1

This worksheet contains a mostly pen and paper calculations. The solutions should be submitted as plain text, files in other formats, like .doc or .pdf will not be accepted. You can write $x\hat{}n$ and a_n for x^n and a_n. For each part of a question write the answer and a short description of how the answer was obtained, so, for example, each part of questions 1 and 2 should have a one sentence description. Each question is worth a fifth of the marks, question 4 and 5 are the extension questions.

1. This question is about estimating the algorithmic complexity of simple operations. Take multiply and addition of a single digit as roughly one operation and assume that powers are calculated by doing the corresponding number of multiplications.

   a) What is the big-oh complexity of adding two n-digit numbers?

   b) What is the big-oh complexity of multiplying two n-digit numbers using the standard long multiplication method.

   c) What is the big-oh complexity of calculating x^n of a single digit number?

   d) What is the big-oh complexity of calculating x^n of an m digit number?

2. This question is about estimating the algorithmic complexity of evaluating a polynomial. Here consider fixed sized variables, so multiplication and addition take roughly one step, irrespective of how many digits the number has. Once again, powers are calculated by multiplication.

   a) What is the big-oh complexity of evaluating, that is finding the value of $p(x)$, of an order n polynomial

   $$p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

   using straight-forward substitution?

   b) Horner's method is a quicker method for evaluating a polynomial. If x_o is the value that the polynomial needs to be evaluated on, let $b_n = a_n$ and then

   $$b_{n-1} = a_{n-1} + x_o b_n$$

   and

   $$b_{n-2} = a_{n-2} + x_0 b_{n-1}$$

   right down to

   $$b_0 = a_0 + x_0 b_1$$

   and $b_0 = p(x_0)$ is the answer. What is the big-oh complexity?

3. This question is intended to help you practice counting things. In the Arthur C. Clarke short story *The Nine Billion Names of God* a 'Mark V Automatic Sequence Computer' is purchase by a religious community to list all the possible names of their deity, a task whose importance is central to their belief system. We are told that this name is nine or fewer characters long and that no characters can occur more than three times in sequence, so AABBAABBA is a possible name, but AAABBAABB is not. We are not told how many characters the alphabet contains. Assuming there are 18 letters and n or few words, how many names are there.

4. Consider another arithmetic operation, or a different algorithm for one of the operations above, describe it and estimate the big-oh complexity. Examples might be long division or the peasant algorithm for multiplication.

5. Consider some other non-sorting algorithm, describe it and estimate the big-oh complexity. Examples might be the Tower of Hanoi or the Traveling Salesman problem.