MCS-375: Algorithms: Analysis and Design Handout #A1
San Skulrattanakulchai
Gustavus Adolphus College Sep 5, 2014

Machine Model & Worst-case Analysis

CLRS: Ch 2.2

Machine Model

A *Random Access Machine (RAM)* is a theoretic model for sequential computers. It has an accumulator, a read-only input tape, a write-only output tape, a program store, and a memory store. It is capable of executing the following instructions:

arithmetic: LOAD, STORE, ADD, SUB, MULT, DIV
control flow: JGTZ, JZERO, JUMP, HALT
input/output: READ, WRITE

Memory is divided into cells, each of which is accessible through a nonnegative integral address. Each memory cell can hold an integer (possibly negative) of arbitrary size. Instructions are not permitted to modify themselves. Both the input and output tapes are unidirectional so the machine cannot go back to re-read/write a previously read/written item. There is an accumulator that can hold an integer of any size. It is by default one of the operands and destination of an instruction.

We write our algorithms in a high-level language, pseudocode, or even English, and mentally translate them into a RAM machine language. For example, $a \leftarrow b+c$ translates to

```
LOAD B
ADD C
STORE A
```

The following examples illustrate some subtleties.

$a \leftarrow (x^2 + y^2)^4$ takes $O(1)$ steps.

$b \leftarrow (x^2 + y^2)^n$, where n is an integer variable, stands for a loop, so it takes $O(n)$ steps.

$c \leftarrow (x^2 + y^2)^{1/2}$ involves calling a square root routine, so its time depends on the number of bits in the desired answer.

Making a procedure call takes $O(1)$ steps.

Time & Space

Two cost criteria are used for RAM programs.

Uniform Cost Criterion Each RAM instruction requires one unit of time; each memory location requires one unit of space.

Logarithmic Cost Criterion The cost of performing an instruction is proportional to the length of the operands of the instructions.

Notes

1. We may augment the instruction set of the RAM to include, for example, bit or character manipulation if convenient.

2. Choosing the appropriate cost criterion to use depends on the problems. For the so-called "number problems" even the logarithmic cost criterion can be an underestimate.

3. Another important theoretical model of sequential computers is the *RASP (Random Access Stored Program) machine*. RAM machines model computers with the Harvard Architecture and RASP machines model those with the von Neumann Architecture. Another important theoretical model is the *Pointer Machine*.

Worst-case Complexity Analysis

The *complexity of an algorithm* is a measure of the amount of computational resources used by the algorithm. Two main resources of interest are time and space. To make the measurement independent of variable parameters like hardware, programming language, computing platform, etc., we measure resource usage asymptotically.

The worst-case time $T(n)$ used by an algorithm is defined as

$T(n) =$ the greatest amount of time spent on a problem of input size n.

Notes

1. Worst-case space function $S(n)$ is defined similar to $T(n)$.

2. We are interested in the *worst-case complexity* of an algorithm more than other kinds of complexity. The reasons are

  (i) It gives bounds on guaranteed performance.

  (ii) It often agrees with practice on real input data.

  (iii) In contrast to the *average-case complexity*, worst-case complexity is mathematically tractable

  (iv) Average case can be unrealistic since the real input distribution is almost never known.

3. *Randomized algorithms* are different from *deterministic algorithms*. They make internal random choices and their worst-case running time don't depend on the probability distribution of the input.

4. A problem that requires reading the input takes time $\Omega(n)$. Therefore, an $O(n)$ algorithm for such a problem is asymptotically optimal.