CS 341 - Algorithms **Winter 2018**

Lecture 2: January 9, 2018

Lecturer: Bin Ma *Notes By: Harsh Mistry*

2.1 Analysing Algorithms

2.1.1 Model of Computation

Model of computation should be both realistic and abstarct. E.g.

- Multiplication slower than addition.Too detailed. Too much distraction fromalgorithm design. In principle, we neglect the difference that causes the increase/reduction of running time by only a constant factor.
- For loop vs. a single arithmetic operation. Should consider. Running time difference affected by input size, more than a constant factor

Word-Ram model

1. Each memory cell is a wordof some bitsthat can hold an integer value.
2. Random access of memory: accessinga word at the location specified by the value of another wordtakes constant time.
3. Basic operations (arithmetic, shifting, logical, comparison) on words take constant time.
4. We further assume thatword is large enough to hold any integer value used in the input.

The word-RAM model closely resemblesthe 32 or 64-bit computers, and is a good balance between reality and simplicity.

2.1.2 Worst Case and Asymptotic Analysis

2.1.2.1 Asymptotical Worst-case Time Complexity

When we analyze the time complexity of an algorithm, we are looking for a simple function $f(n)$, such that the running time of the algorithm for an input size n is $O(f(n))$

$O(f(n))$ is also called the asymptotic **Upper Bound** of the algorithm's running time. If for some constant c, we can find infinity number of instances such that the time complexity is at least $c \times f(n)$, then we say that bounf $O(f(n))$ is tight.

Speaking of the worst-case time complexity of an algorithm, it is conventional to write a tight upper bound on the running time $O(f(n))$ as $\Theta(f(n))$

2.1.2.2 Average Case

Average case assumes a probabilistic distribution, and calculate the weighted average of the running time.

This makes sense in some situations. E.g. Hash table. Also, if we know that many of the instances our sorting algorithm encounters are actually already sorted, then the earlier Smart sort algorithm is actually a good idea. In fact, one of the practically used sorting algorithm, Tim Sort, makes use of the property that the array may have a lot of already sorted regions in practical examples.But in CS 340, if not otherwise specified, we always look for worst-case time complexity

2.1.3 Reduction

Reduction is a useful technique in algorithm design where if you already have algorithm for problem X, and you need design an algorithm for problem Y. Instead of designing a new algorithm, you design a strategy to convert an instance of I_Y of Y to an instance of I_X of X; then solve I_X with the algorithm for X; then you convert teh solution back.

In essence, reduction is the process of utilizing a given algorithm to solve a different problem by transforming the output or chaining algorithms.