CS 430 Introduction to Algorithms – Fall 2017

Recitation Class Note 1

Xiaolang Wang

1. Asymptotic Analysis

* Running time: on a particular input, the number of primitive operations or “steps”
* Rate of growth
* Asymptotic Rate – use only the term that dominate the rate of growth, or think of it like the bottleneck of an algorithm.
* Worst case, Best case, average case
* Ω() – asymptotically lower bound, Θ() – asymptotically tight bound, O() – asymptotically upper bound, ω(), o().
* Worst case ≠ O()

Questions:

1. True or false

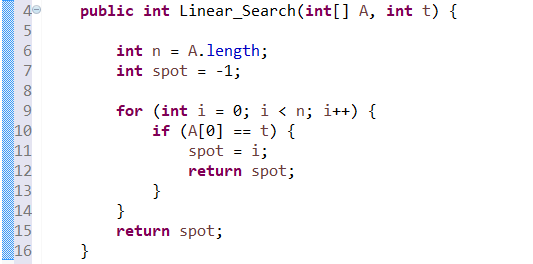
* Given and, we have
* If , then

1. Order the following lists of functions by their big-theta notations. Also, mark all functions of the same growth rate.

* TIP: exponential > polynomial > logarithmic

1. Linear Search

Worst case? Best case? Average case?



1. Some requirements for homework

* Answer exactly what is asked in question, a long answer might not be a correct answer.
* If you solve a question in multiple ways (and the question doesn’t ask you to), only the first solution will be checked.
* To give an algorithm:

1. You can use pseudo code, and be careful of “layers” (like, where is the start and the end of a loop) and other syntax/semantic mistakes.
2. You can use algorithms/operations already given in the textbook: for example, if you need your input to be sorted inside of your algorithm, you can just write QuickSort(1…n) in that step instead of the sorting details.
3. You can use words to describe your algorithm too; make sure your sentences are clear and the operations you use are not vague.

* To analyze the rate of growth: any specific input (an example) will be considered as loss of generality; so focus on the general case. For example, when analyzing the time complexity of insertion sort, some students use an example (maybe 2, 7, 9, 1, and 4) to demonstrate what happens in the function; this is bad.
* To prove:

1. You can try induction (prove base case, then prove the “k+1th” case to be true with the assumption that the “kth” case is true), construction (construct or find something when you want to prove it exists), and contradiction (if the statement is wrong, then something bad will happen; thus the statement is true).
2. Don’t skip any logic steps. For example, when you are asked to prove k > 1, and after you proved k > 2, don’t forget to say “because 2 > 1, so k > 1”.