Asymptotics Drill 2

Tom Bohbot

Octover 2020

1 Comparing Relative Order-of-Growth Of Two Functions

Row	f(n)	g(n)	Your Answer (1/2/3)
1	log n^2	logn + 5	3
2	$\sqrt{n}$	log n^2	2
3	$\log^2$ n	logn	2
4	n	$\log^2$n	2
5	nlogn + n	logn	2
6	10^{10}	log 42	3
7	2^n	10n^2	2
8	2^n	1.5^n	2
9	2(logn)^2	logn + 1	2
10	2^n	2^{2n}	1

2 Asymptotics

2.1

You've proven that an algorithm is O(n^2): can I infer that it cannot take O(n) on some inputs?
No, it is possible for an alogirthm to take O(n) time on some inputs. For example, Insertion sort takes O(n^2) time worst case, but if a sorted listed is inserted it will only take O(n) time to run.

2.2

You've proven that an algorithm takes O(n^2): can I infer that it takes O(n) on all inputs?
No, Being O(n^2) means that the algorithm can take up to n^2 time to run, so one should not assume that it will only take O(n) time to run.

2.3

You've proven that an algorithm takes $\Theta(\mathrm{n}^2)$ worst-case time: can I infer that it cannot take $\mathrm{O(n)}$ on some inputs?
No, if an algorithm takes $\mathrm{O(n^2)}$ time, this means that Big-Omega and Big-O are both $\mathrm{O(n^2)}$, which proves that this algorithm can never be $\mathrm{O(n)}$. However, if inputs are ideal then the running time of the algorithm can decrease like in insertion sort going from n^2 to n when the list inserted is pre-sorted.

2.4

You've proven that an algorithm takes $\Theta(\mathrm{n}^2)$: can I infer that that it's impossible for it to take $\mathrm{O(n)}$ on all inputs?
Yes, one can infer this because $\Theta(\mathrm{n}^2)$ means that the algorithm will take n^2 worst case, which means that at least one case disproves this statement.

2.5

Given a function where $\mathrm{f(n)} = 100\mathrm{n}^2$ for even n and $\mathrm{f(n)} = 20\mathrm{n}^2$ (n log2n) for odd n, can we claim that $\mathrm{f(n)} = \Theta(\mathrm{n}^2)$?
Yes, since only the leading terms are considered when calculating Θ runtime and both of these functions are just $\Theta(\mathrm{n}^2)$, this algorithm has a $\Theta(\mathrm{n}^2)$ runtime.