Algorithms, upper/lower bounds, and best/worst case [duplicate]

This question already has an answer here:
Upper bound vs lower bound for worst case running time of an algorithm *2 answers*

For algorithms, how are the bounds related to best/worst cases? Is the worst case synonymous with upper bound, and best case synonymous with lower bound? Or can you at least derive one from the other? Or are they not related at all?

algorithm lower-bound upperbound

asked Jun 21 '15 at 16:55
K-Smoove
6 3

marked as duplicate by shauryachats, Juan Lopes, beaker, David Eisenstat algorithm Jun 25 '15 at 17:50

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

2 Answers

Yes it can say that worst case synonymous with upper bound and best case synonymous with lower bound. Worst-case performance are the most used in algorithm analysis.In the worst analysis, we guarantee an upper bound on the running time of an algorithm which is good information.In the worst case analysis, we calculate upper bound on running time of an algorithm. We must know the case that causes maximum number of operations to be executed. While in the best case analysis, we calculate lower bound on running time of an algorithm. We must know the case that causes minimum number of operations to be executed.

Yes we can drive the relationship between wrost case and best case with the help of average case analysis. Let O, Θ, Ω represent the wrost case, average case and best case respectively and f(n) and g(n) are two functions. 1) If f(n) = O(g(n)) and f(n) = Θ(g(n)) ==> f(n) = Ω(g(n))
2) If f(n) = Ω(g(n)) and f(n) = Θ(g(n)) ==> f(n) = O(g(n))

answered Jun 21 '15 at 17:47

rishi kant
651 2 20

In the worst case analysis, we calculate upper bound on running time of an algorithm. We must know the case that causes maximum number of operations to be executed. For Linear Search, the worst case happens when the element to be searched (x in the above code) is not present in the array.*Most of the times, we do worst case analysis to analyze algorithms. In the worst analysis, we guarantee an upper bound on the running time of an algorithm which is good information.*

In the best case analysis, we calculate lower bound on running time of an algorithm. We must know the case that causes minimum number of operations to be executed. In the linear search problem, the best case occurs when x is present at the first location.*The Best Case analysis is bogus. Guaranteeing a lower bound on an algorithm doesn't provide any information as in the worst case, an algorithm may take years to run.*

answered Jun 21 '15 at 17:07

Deepak Tiwari
259 1 17

I realize that the best case scenario is generally not very useful, but I read that the lower bound of an algorithm is important b/c it proves that no algorithm can do better. So does this mean that lower bound and best case are two different things? – K-Smoove Jun 21 '15 at 17:20

A typical lower bound proof says that any algorithm to solve a problem must take at least time f(n) - e.g. any sort using only comparisons must make at least n lg n comparisons in the worst case. This is not the same as a best case scenario which might say, for example, that a bubble sort may in the best case only make n comparisons. – mcdowella Jun 21 '15 at 17:26

@K-Smoove best case is synonymous to lower bound, or I would say that in best case analysis we will consider the lower bound on running time. – Deepak Tiwari Jun 22 '15 at 17:05