

Algorithm Efficiency Problem

- For each of the following functions, indicate how much the function's value will change if its argument is increased eightfold. Use either (i) the difference between, or (ii) the ratio of, $f(8n)$ and $f(n)$, whichever is more convenient for getting a compact answer. Note that none of the answers depend on n .
 - $f(n) = \lg n$
 - $f(n) = \sqrt[3]{n}$
 - $f(n) = n$
 - $f(n) = n^2$
 - $f(n) = n^3$
- Consider a variation of sequential search that scans a list to return the number of occurrences of a given search key in the list. Does its efficiency differ from the efficiency of classic sequential search? Your answer should include three parts, one each for the Best Case, Average Case, and Worst Case time efficiencies. Explain your answers.
- Describe how to implement a constant-time operation on an unsorted list of size n that deletes the i^{th} element ($0 \leq i < n$). Remember, the running time of this operation must be constant; in other words the running time must not depend on the number of items in the list.
- Algorithm \mathcal{A} performs $966n^2$ basic operations, and algorithm \mathcal{B} performs $24n^3$ basic operations. For what value of n does algorithm \mathcal{A} start to show its better performance?
- There are two algorithms called *Algo 1* and *Algo 2*. For an input size of n , *Algo 1* runs in n^3 microseconds and *Algo 2* runs in $200n \log n$ microseconds. *Algo 1* can be implemented using 6 hours of programmer time and 7 minutes of CPU time. On the other hand, *Algo 2* requires 22 hours of programmer time and 12 minutes of CPU time to implement. If programmers are paid \$25 per hour and CPU time costs \$60 per minute, how many times must a problem of size $n = 600$ be solved using *Algo 2* to justify its development cost?