

1 MergeSort

See the source code file `mergeSort.cpp` and the tests given in `tests.cpp`.

2 Big-O Proofs

Problem 1. Show that $6n^2n + 4$ is $O(n^2)$.

Proof. Let $f(n) = 6n^2 - n + 4$. We want to find constants c and n_0 such that for all $n > n_0$,

$$f(n) < c \cdot n^2.$$

Let $c = 7$ and $n_0 = 10$. Consider the function $f(n) - c \cdot n^2 = 6n^2 - n + 4 - 7n^2 = -n^2 - n + 4$. For all $n > n_0$, $f(n) - c \cdot n^2 = -n^2 - n + 4 < 0$. Therefore for the given values of n_0 and c , $f(n) - c \cdot n^2 < 0$ for all $n > n_0$, which is equivalent to saying:

$$f(n) < c \cdot n^2, \tag{1}$$

for all $n > n_0$ as desired. \square

Problem 2. Show that $n^3 + 8n^2 - 12$ is $O(n^3)$.

Proof. Let $c = 10$ and $n_0 = 10$. $f(n) = n^3 + 8n^2 - 12$ and $c \cdot n^3 = 10n^3$. For all $n > n_0$, $f(n) - cn^3 = -9n^3 + 8n^2 - 12 < 0$. Therefore:

$$f(n) < c \cdot n^3, \tag{2}$$

for all $n > n_0$ as desired. \square

3 Mystery Functions

General approach: For high values of n , the power of an upper bound function is the dominant factor in the difference in runtime with respect to other functions. First, we are going to list the least upper bound functions for each of the lettered functions.

3.1 Obtaining the upper bounds.

- A The for loop is iterated n times, with one operation per iteration. Therefore $\hat{O}(fnA(n)) = n$.
- B The for loop is iterated $4n$ times, with one operation per iteration. Therefore $\hat{O}(fnB(n)) = 4n$.
- C The inner 'j' for loop is iterated n times; the outer 'i' for loop is iterated n times as well. Therefore $\hat{O}(fnC(n)) = n^2$.
- D The for loop is iterated n^3 times, with two operations per iteration. Therefore $\hat{O}(fnD(n)) = 2n^3$.
- E The inner 'j' for loop is iterated n^2 times; the outer 'i' for loop is iterated n^2 times as well. Therefore $\hat{O}(fnE(n)) = n^2 \cdot n^2 = n^4$.
- F The inner 'j' while loop is iterated 10 times, with two operations per iteration; the outer 'i' for loop is iterated n^2 times. Therefore $\hat{O}(fnF(n)) = 20 \cdot n^2 = 20n^2$.

function letter	upper bound
A	n
B	$4n$
C	n^2
D	n^3
E	n^4
F	$20 \cdot n^2$

Figure 1: For convenience, the table of upper bounds for the lettered functions.

3.2 Comparisons and conclusions.

Referring to figure 2 below, we can see that with all functions plotted, $f3$ has the longest runtime against other functions especially at large values of n . Therefore the upper bound of $f3$ has the highest power. According to figure 1, mystery function $f3$ is fE which has the highest power.

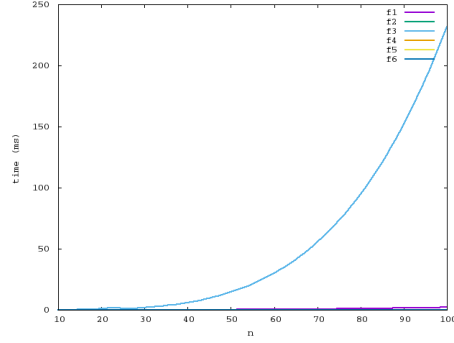


Figure 2: Runtime vs n with $n \in [10, 100]$.

Referring to figure 3 below, excluding mystery function $f3$, $f1$ has the longest runtime against the remaining functions especially at large values of n . Therefore the upper bound of $f1$ has the highest power among the remaining mystery functions. According to figure 1, mystery function $f1$ is fD which has the highest power.

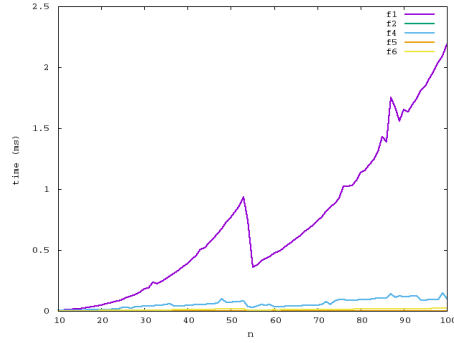


Figure 3: Runtime vs n with $n \in [10, 100]$.

Referring to figure 4 below, excluding mystery functions $f3$ and $f1$, $f4$ and $f6$ have significantly longer runtime against the remaining functions especially at large values of n . Therefore the upper bound of $f4$ and $f6$ have the highest power among the remaining mystery functions. Furthermore, it is very likely that the upper bounds of $f4$ and $f6$ are off by only a constant of proportionality. Therefore we can conclude that according to figure 1, $f4$ which has a larger constant of proportionality is fF and $f6$ is fC .

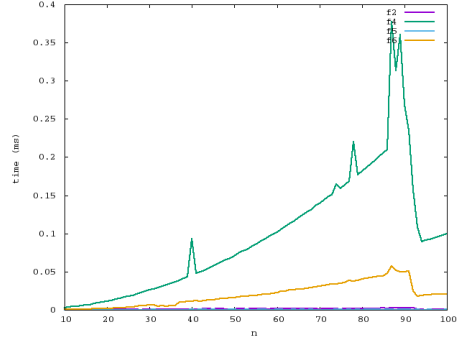


Figure 4: Runtime vs n with $n \in [10, 100]$.

Referring to figure 1, the upper bounds of $f2$ and $f5$, by process of elimination should have the lowest power. According to figure 5, the functional behavior of runtime vs n is similar between $f2$ and $f5$. Thus $f2$ should be fB which has a higher constant of proportionality. And similarly $f5$ should be fA .

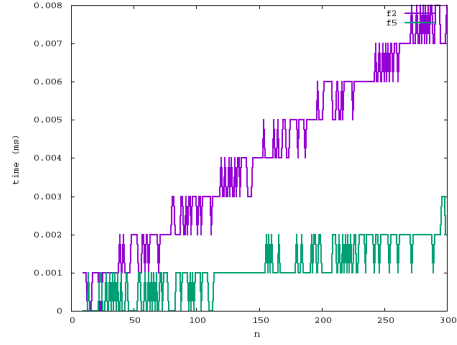


Figure 5: Runtime vs n with $n \in [10, 100]$.

Below is a compiled table of results:

mystery function	corresponding lettered function
$f1$	fD
$f2$	fB
$f3$	fE
$f4$	fF
$f5$	fA
$f6$	fC

Figure 6: Table of results.