CSE 8833
HW Assignment #1
09/11/2023
Group #1

Omid Abdolazimi
Adam Thomas
Charles Albright
Joseph Jabour
Martin Duclos
Robert Ward

Problem 1

In the figure below (**Figure 1**), $f_1(n)$ and $f_2(n)$ represent the running times of two algorithms for the same problem. The complexity of the two algorithms is represented by the asymptotic tight bounds $g_1(n)$ and $g_2(n)$ of these running times with respect to the input size *n*.

$$\Theta(g_1(n)) = \left\{ \begin{array}{l} f_1(n) : \exists \text{ positive constants } c_1,\ c_2,\ n_{01} \text{ such that} \\ 0 \le c_1 g_1(n) \le f_1(n) \le c_2 g_1(n),\ n \ge n_{01} \end{array} \right\}$$

$$\Theta(g_2(n)) = \left\{ \begin{array}{l} f_2(n) : \exists \text{ positive constants } c_3,\ c_4,\ n_{02} \text{ such that} \\ 0 \le c_4 g_2(n) \le f_2(n) \le c_3 g_2(n),\ n \ge n_{02} \end{array} \right\}$$

Discuss the behavior of these two algorithms for values of *n* ranging from 0 to ∞.

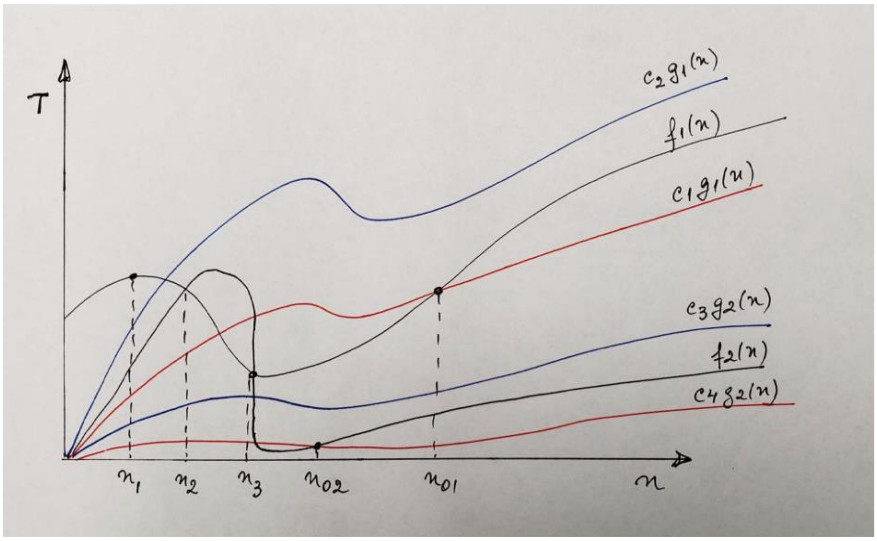

Figure 1. Running time vs input size for two arbitrary algorithms f_1 and f_2

Answer:

Let algorithm *A*'s running time be represented by the function $f_1(n)$ with input size *n* and algorithm *B*'s running time be represented by the function $f_2(n)$ with input size *n*. It is given that $f_1(n) = \Theta(g_1(n))$ and $f_2(n) = \Theta(g_2(n))$, thus there exist c_1, c_2, n_{01} in *N* such that $0 \le c_1 g_1(n) \le f_1(n) \le c_2 g_1(n)$, $n \ge n_{01}$ and there exist c_3, c_4, n_{02} in *N* such that $0 \le c_4 g_2(n) \le f_2(n) \le c_3 g_2(n)$, $n \ge n_{02}$.

In the graph provided, in [0, n_1), $f_1(n)$ starts at some time value $T(n) \succ 0$. Perhaps algorithm *A* begins with a large data ingest. Then, while in [0, n_1), $f_1(n)$, and $f_2(n)$ are both increasing. So, the workload of both algorithm *A* and algorithm *B* are increasing. Also note that while in [0,

Omid Abdolazimi
Adam Thomas
Charles Albright
Joseph Jabour
Martin Duclos
Robert Ward

1), $f_2(n) \succ c_3 g_2(n)$ and $f_1(n) \succ c_2 g_1(n)$. However, since $n_1 \leq n_{02} \prec n_{01}$ and $n_1 \prec n_{02}$, that is not a problem for our asymptotic tight bounds.

To continue along the horizontal axis, for [n, n_2), $f_1(n)$ is decreasing, implying algorithm *A*'s workload is lessening. $f_2(n)$ is still increasing, implying algorithm *B*'s workload is increasing. In [n, n_2), $f_1(n)$ intersects with $c_2 g_1(n)$ and $f_2(n) \succ c_3 g_2(n)$, but again, as $n_2 \prec n_{01}$ and $n_2 \prec n_{01}$, this is not a problem for our asymptotic tight bounds.

Continuing, for [n_2, n_3), $f_1(n)$ is dramatically decreasing, implying a decline in algorithm *A*'s workload. $f_2(n)$ is increasing, then begins a decrease, and finally a sharp plummet, implying the start of a sharp drop in workload at n_3 for algorithm *B*. In [n_2, n_3), $f_1(n)$ intersects with $c_1 g_1(n)$ and $f_2(n) \succ c_3 g_2(n)$, but as $n_3 \prec n_{01}$ and $n_3 \prec n_{02}$, this is not a problem for our asymptotic tight bounds.

For [n_3, n_{02}), $f_1(n)$ is increasing, implying algorithm *A*'s workload is increasing. $f_2(n)$ completes its sharp drop, then "levels out," so one could conclude algorithm *B* reaches a steady behavior. In [n_3, n_{02}), $f_1(n) \prec c_1 g_1(n)$ and $f_2(n)$ intersects with $c_3 g_2(n)$ once and $c_4 g_2(n)$ twice. In both cases, we are still at $n_1 \leq n_{02} \prec n_{01}$, so this is not a problem for our asymptotic tight bounds.

For [n_{02}, n_{01}), $f_1(n)$ is increasing, implying algorithm *A* has an increasing workload. $f_2(n)$ is barely increasing, implying a slightly increasing workload. In [n_{02}, n_{01}), $f_1(n) \prec c_1 g_1(n)$, so we still have not satisfied its asymptotic bounds, but that is ok because we are at $n_1 \geq n_{01}$. However, we now have that $0 \leq c_4 g_2(n) \leq f_2(n) \leq c_3 g_2(n)$, for all $n_1 \geq n_{02}$. So the value of $f_2(n)$ always lies between $c_4 g_2(n)$ and $c_3 g_2(n)$, inclusive, at and to the right of n_{02}. $c_3 g_2(n)$ provides an upper bound for $f_2(n)$ and $c_4 g_2(n)$ provides a lower bound for $f_2(n)$. Algorithm *B* follows the same growth rate as other algorithms represented by functions in $\Theta(g_2(n))$.

Finally, in [n_{01}, n), $f_1(n)$ increases and $f_2(n)$ slightly increases, implying both algorithms *A* and *B* have workloads that increase. In [n_0, 1), we now have that that $0 \leq c_1 g_1(n) \leq f_1(n) \leq c_2 g_1(n)$, for all $n \geq n_{01}$. So the value of $f_1(n)$ always lies between $c_1 g_1(n)$ and $c_2 g_1(n)$, inclusive, at and to the right of n_{01}. $c_2 g_1(n)$ provides an upper bound for $f_1(n)$ and $c_1 g_1(n)$ provides a lower bound for $f_1(n)$. Algorithm *A* follows the same growth rate as other algorithms represented by functions in $\Theta(g_1(n))$.

Overall, algorithm *B* [related to $f_2(n)$] outperforms algorithm *A* [corresponding to $f_1(n)$] for all *n* except [n_2, n_3].

CSE 8833
HW Assignment #1
09/11/2023
Group #1

Omid Abdolazimi
Adam Thomas
Charles Albright
Joseph Jabour
Martin Duclos
Robert Ward

Problem 2

Evaluate the following statements for correctness. Answer each of them with 'true' (**T**) or 'false (**F**):

Answer:

a) $n^3 = \Omega(n^2)$ **True**

***Proof*:**

Show there exist c, n_0 in N such that $f(n) \geq cg(n)$ for all $n \geq n_0$.

$n^3 \geq cn^2$

Pick $c = 1$, $n_0 = 1$. $n^3 \geq n^2$ for all $n \geq 1$.

---

b) $n + \log n = O(n)$ **True**

***Proof*:**

Show there exist c, n_0 in N such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

$0 \leq n + \log n \leq cn$ ➔ $\log n \leq cn - n$

$\log n \leq n(c-1)$ ➔ $n \leq 10^{n(c-1)}$

Pick $c = 2$, $n_0 = 1$. $n \leq 10^n$ for all $n \geq 1$.

---

c) $n + n\log n = O(n)$ **False**

***Proof*:**

Show there exist c, n_0 in N such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

$0 \leq n + n\log n \leq cn$ ➔ $n\log n \leq cn - n$

$n\log n \leq n(c-1)$ ➔ $\log n \leq c - 1$

---

d) $n^2 \log n = \theta(n)$ **False**

***Proof*:**

Show there exist c_1, c_2, and n_0 in N such that $c_1 g_1(n) \leq f(n) \leq c_2 g_2(n)$ for all $n \geq n_0$.

$c_1 n^2 \leq n^2 \log n \leq c_2 n^2$ ➔ $c_1 \leq \log n \leq c_2$

---

e) $n! = O(2^n)$ **False**

***Proof*:**

Show there exist c, n_0 in N such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

$0 \leq n! \leq c2^n$

$n \times (n-1) \times (n-2) \times \ldots \times (2) \times (1) \leq c \times 2 \times 2 \times \ldots \times 2 \times 2$

n numbers on left, n numbers on right.

Note: $n \succ 2$, $(n-1) \succ 2$,... until 2.