# CS412 Algorithms: Design & Analysis

Spring 2024

**Dhanani School of Science and Engineering**
Habib University

Copyright © 2024 Habib University

Practice Problems

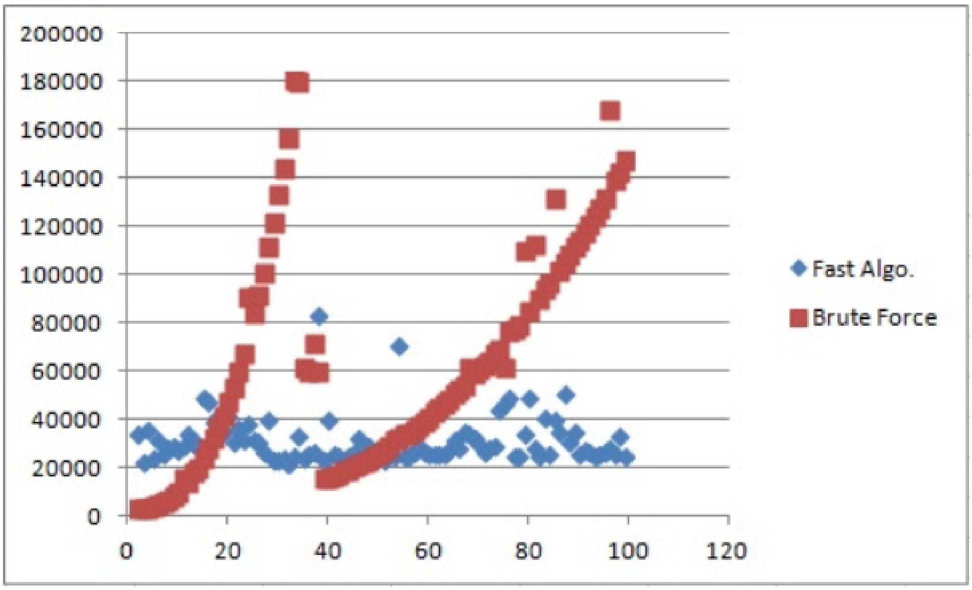
Week 5 & 6

1. What does FIND-MAXIMUM-SUBARRAY return when all elements of A are negative?

**Solution:** It will return the least negative position. As each of the cross sums are computed, the most positive one must have the shortest possible lengths. The algorithm doesn't consider length zero sub arrays, so it must have length 1.

2. Implement both the brute-force and recursive algorithms for the maximumsubarray problem on your own computer. What problem size $n_0$ gives the crossover point at which the recursive algorithm beats the brute-force algorithm? Then, change the base case of the recursive algorithm to use the brute-force algorithm whenever the problem size is less than $n_0$. Does that change the crossover point?

**Solution:** The crossover point is at around a length 20 array, however, the times were incredibly noisy and I think that there was a garbage collection during the run, so it is not reliable. It would probably be more effective to use an actual profiler for measuring runtimes. By switching over the way the recursive algorithm handles the base case, the recursive algorithm is now better for smaller values of $n$. The chart included has really strange runtimes for the brute force algorithm. These times were obtained on a i7 10750H.
In the chart of runtimes, the $x$ axis is the length of the array input. The $y$ axis is the measured runtime in nanoseconds.




3. Show that the solution to $T(n) = 2T(\lfloor n/2 \rfloor + 17)$ is $O(n\lg(n))$.

**Solution:** Choose $n_1$ such that $n \geq n_1$ implies $n/2 + 17 \leq 3n/4$. We'll find $c$ and $d$ such that $T(n) \leq cn\text{log}n - d$.

$$T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$$

$$\leq 2(c(n/2 + 17)\text{log}(n/2 + 17) - d) + n$$
$$\leq cn\text{log}(n/2 + 17) + 17c\text{log}(n/2 + 17) - 2d + n$$
$$\leq cn\text{log}(3n/4) + 17c\text{log}(3n/4) - d + n$$
$$= cn\text{log}n + cn\text{log}(3/4) + 17c\text{log}(3n/4) - 2d + n$$

Take c $= -2/\text{log}(3/4)$ and $d = 34$. Then we have $T(n) \leq cn\text{log}n - d + 17c\text{log}(n) - n$. Since $\text{log}(n) = o(n)$, there exists $n_2$ such that $ngeqn_2$ implies $n \geq 17c\text{log}(n)$. Letting $n_0 = max\{n_1, n_2\}$ we have that $n \geq n_0$ implies $T(n) \leq cn\text{log}n - d$. Therefore $T(n) = O(n\text{log}n)$.

4. Using the master method, you can show that the solution to the recurrence $T(n) = 4T(n/2) + n^2$ is $T(n) = \Theta(n^2)$. Show that a substitution proof with the assumption $T(n) \leq cn^2$ fails. Then show how to subtract off a lower-order term to make a substitution proof work.

**Solution:**

Suppose we want to use substitution to show $T(n) \leq cn^2$ for some $c$. Then we have

$$T(n) = 4T(n/2) + n$$

$$\leq 4(c(n/2)2) + n$$
$$= cn2 + n,$$

which fails to be less than $cn^2$ for any $c > 0$. Next we'll attempt to show $T(n) \leq cn^2 - n$.

$$T(n) = 4T(n/2) + n$$

$$\leq 4(c(n/2)2 - n) + n$$
$$= cn^2 - 4cn + n$$
$$\leq cn^2$$