Midterm Project - Brute Force vs Divide and Conquer
Spring 2017

In this project we want to look at different problems and compare a Brute Force approach with a Divide and Conquer (D&C) approach.

---

1. In this problem we have a list of numbers such as

$$5, 15, 3, 11, 9, 12, 4, 8, 10, 19, 2, 6$$

and we want to put them in ascending order, i.e., create the new list

$$2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15, 19$$

This is called sorting.

a. A Brute Force approach might create a new sorted list by going through the list (assume it has N numbers in it) and finding the smallest number and move it to the first position in the new list. Then it goes through the $N - 1$ remaining numbers and finds the smallest number in that list and then puts it in the second position. Now there are $N - 2$ numbers which have to be sorted. This process continues until there is only one number left which, of course, doesn't have to be sorted.

For the given list of 12 numbers complete a table like the one below.

entry # in sorted list	# of comparisons required
1	11
2	
3	
⋮	
12	

What is the total number of comparisons required? Recall that there is a formula to determine the sum of integers like $1 + 2 + 3 + \cdots + N$ which is given by

$$1 + 2 + 3 + \cdots + N = \frac{(N+1)(N)}{2}$$

How many comparisons are needed for a list of 10, 20, 40, 80, 100 numbers? Plot your results where N is on the $x-$ axis and the number of steps on the y-axis.

b. One D&C approach to sorting is called QuickSort. We will describe it in terms of our example.

First we choose a pivot element in the list which is typically the right most entry, in our case 6. Now we divide the array and put everything < 6 to its left and everything $>= 6$ to its right. These entries are in no special order. For example, we might have 5, 3, 4, 2, 6, 15,11, 9,12, 8, 10, 19 after the first divide step. Now we have reduced the problem to two smaller lists. We note that the pivot number (6) is in position 5 in the list. Now we recursively sort the list to the left of position 6, i.e., in positions 1-5 and sort the list starting in position 7 and ending in 12. To sort each of these 2 lists we use the D&C approach. For example, for the list 5, 3, 4, 2 we choose the pivot element as 2 and sort it into 2 smaller lists 2 and 5,3,4. The first of these only has one element so we don't have to do anything. We divide the second list by choosing 4 as the pivot and sort into two lists the first consisting of 3 and the second 5. We see that each list only has 1 element so we stop. Now we know that the first 3 elements are 3,4,5. We then return to the list 15, 11, 9, 12, 8, 10, 19 and use D&C to sort it.

i. Determine the number of comparisons that must be made to sort our array using Quicksort and compare with the Brute Force approach.

ii. Go through the steps of the D&C approach to sort the list

$$5, 9, 2, 21, 17, 4, 6, 55, 33, 15, 71, 11, 107, 7, 14$$

---

2. We can extend the Brute Force and D&C ideas in (1) to search an **ordered** list. Suppose we are searching the list

$$2, 4, 5, 6, 7, 9, 14, 15, 17, 21, 29, 33, 55, 71, 107$$

for the number 71.

a. Describe the Brute Force approach for searching the given list for 71. If you have a list of length N, what is the maximum number of steps that you have to do? Explain your result.

b. Describe an extension of the Quicksort routine to search the ordered list above. Go through the steps in your algorithm to locate 71.

c. Suppose you have a random list of N numbers (not ordered) which you want to search many times for different numbers. Combine the ideas in this problem and the previous problem to devise a D&C algorithm to do this.

---

3. In this problem we want to compute a given number to a large, even power; for example compute $y = 5^{32}$.

a. The Brute Force approach is to just multipy the number times itself the desired number of times. Here is an algorithm for computing $y = x^n$ when x is known using the Brute Force approach.

Set $y = 1$.
For $k = 1, 2, 3, \ldots, n$
    $y = y \times x$
end loop

Suppose we want to determine $y = 5^4$ using this algorithm. Make a table of k and the value of y at the end of each step. How many multiplications does it take? If we want to compute $y = 5^{64}$, how many multiplications does it take? Determine the number of steps for computing 5^n when $n = 2, 4, 8, 16, 24, 32, 64$. Then plot your results where n is on the $x-$ axis and the number of steps on the y-axis.

b. For the D&C algorithm for finding 5^{16} we take advantage of how exponents are computed. For example, if we have $2^2 \cdot 2^3 = 4 \times 8 = 32$ then this is the same thing as 2^5 where we have added the exponents 2 and 3 in the expression $2^2 \cdot 2^3$. This means that if the exponent n is even then

$$5^n = 5^{n/2} 5^{n/2} .$$

So for 5^{16} we write it as $5^{16} = 5^8 5^8$ and thus reduce our problem of finding 5^{16} to finding 5^8 and multiplying it by itself. Now we divide again to get $5^8 = 5^4 5^4$. If we know 5^4 then we square it to get 5^8 and square this result to get 5^{16}. Next we divide again to get $5^4 = 5^2 \cdot 5^2$ and lastly to get $5^2 = 5^1 \cdot 5^1$. Now how do we combine our results to get the answer? We compute 5^2 which requires one multiplication; then we use this result to compute $5^4 = 5^2 \cdot 5^2$ which is another multiplication. Now $5^8 = 5^4 \cdot 5^4$ can be computed with another multiplication. To get 5^{16} we just multiply 5^8 times itself so we have computed 5^{16} with a total of 4 multiplications instead of 15 which is required by the Brute Force approach. Here we are assuming that all powers are even. In (c) you are asked to go through an example when a power is odd.

(i) Go through the steps for computing 5^{32} using the D & C approach. How many multiplications does it take?

(ii) How many multiplications will it take to compute 5^{64}, 5^{128}, 5^{128} and 5^{256}. Make a plot with the power n on the x-axis and the number of multiplications on the y-axis.

c. Discuss how you would modify the D&C algorithm in (b) to compute 5^{20}.
