Midterm Algorithm Spring 2025

Do five out of six problems!

Name: _____

Sign: I hereby recognize that I should not submit more than 5 solutions to problems.

_____

Master Theorem (abridged):

Given $T(n) = aT(n/b) + f(n)$. Set $c = \log_b(a)$.
Case 1: If $f(n) = O(n^{c-\epsilon})$ for some $\epsilon > 0$, then $T(n) = \Theta(n^c)$.
Case 2: If $f(n) = \Theta(n^c)$, then $T(n) = \Theta(n^c \log(n))$.
Case 3: If $f(n) = \Omega(n^{c+\epsilon})$ for some $\epsilon > 0$ (and $a(f(n/b) \leq Cf(n)$ for some $C < 1$ eventually), then $T(n) = \Theta(f(n))$.

Problem 1:

Determine the asymptotic runtime of the following code in dependence on *n:*

```
def sort(array):
      n = len(array)              #this takes constant time
      if n < 10:
            bubble_sort(array)    #sorts the array in place
      a = n//5
      b = 2*n//5
      c = 3*n//5
      d = 4*n//5
      sort( array[0:c] )
      sort( array[a:d] )
      sort( array[b:n] )
      sort( array[0:c] )
      sort( array[a:d] )
      sort( array[b:n] )
```

Your argument does not have to count with rounding errors, e.g. you can assume that the first recursive call to sort involves an array with $3n/5$ elements.

Problem 2:

You are given an array consisting only of single characters 'b', 'w', and 'r'. Find a linear time algorithm to order them so that they make up the tricolor, where all 'b' are before all 'w' which are before all 'r'. For example, given the array ['w', 'w', 'b', 'b', 'w', 'r', 'b', 'w','r'], the result should be ['b', 'b', 'b', 'w', 'w', 'w', 'w','r', 'r'].

10% Extra Credit

Problem 3:

A *contiguous* sub-array of an array is an array consisting of neighboring elements. Thus, [3,4,5] is a contiguous sub-array of [1,2,3,4,5,6,7], but [1,2,4] is not contiguous. Give a linear time algorithm that determines the maximum sum of a contiguous sub-array.

Hint: find `MSCS(i)`, the maximum sum of a contiguous sub-array of the first i elements. To do so, find `MLR(i)`, the maximum sum of a contiguous sub-array of the first i elements that ends with the i-th element.

You can check your algorithm with the following test cases:

```
[2, 4, -10, 3, 4, -5, 4, 3] 9
[1, 2, 3, -10] 6
[2, 5, -3, 1, -20, -3, 2, 3] 7
```

Problem 4:

Assume you are involved in the design of a very small micro-processor. Unlike modern micro-processors, the contemplated design does not have dedicated floating-point multipliers but uses instead micro-code so that multiplying floating point numbers takes disproportionate time. A design team proposes to include a unit that squares floating-point numbers in hard-ware, as they found a way to do so effectively in a single clock cycle. The team argues that squaring is the most important type of floating point multiplication, but receive criticism from the rest who argue that this is not the case. Since you are the only one in the company who took an Algorithm class, you are asked to answer the following question:

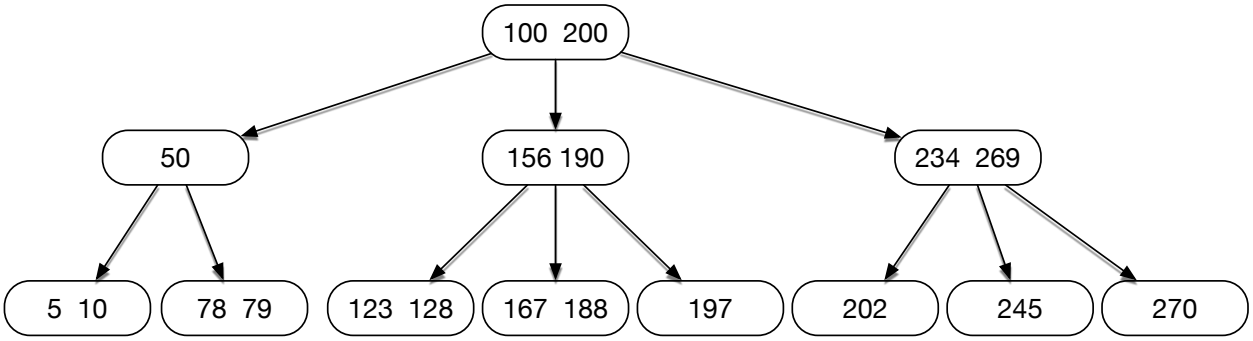
Can we speed up floating point multiplication if we have an efficient squaring method?

Hint: They already told you that they can quickly multiply or divide by powers of two.

Problem 5:

An LH-file has 19 buckets. What is the level and what is the split-pointer. To which buckets do the three elements with key-hashes 33, 40, and 49 go. Show your calculation.

Problem 6:



In the above 2-3 (B-)tree, where only the key hashes are shown, we are inserting a record with key-hash 125. Explain who the insertion function, giving all stages of restructuring.