Midterm Algorithms Fall 2024

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:

Give a recurrence relation for the following (bogus)-algorithm that operates on an array of n elements. Your recurrence relation does not need to reflect the rounding up or down of fractions.

```
def sal(array):
    n = len(array)
    if n == 1: return
    elif n == 2:
        array[0], array[1] = min(array), max(array)
        return
    else:
        sal(array[0 : 2*n//3])
        sal(array[n//3 : n])
        sal(array[0 : 2*n//3])
        sal(array[n//3 : n])
        return
```

Then use the Master Theorem (if possible) to determine the approximate run-time.

Problem 2:

You are given an array of words such as

```
['We', 'therefore', 'the', 'Representatives', 'of', 'the', 'united', 'States', 'of', 'America',
'in', 'General', 'Congress', 'Assembled', 'appealing', 'to', 'the', 'Supreme', 'Judge', 'of',
'the', 'world', 'for', 'the', 'rectitude', 'of', 'our', 'intentions', 'do', 'in', 'the', 'Name',
'and', 'by', 'Authority', 'of', 'the', 'good', 'People', 'of', 'these', 'Colonies', 'solemnly',
'publish', 'and', 'declare', 'That', 'these', 'United', 'Colonies', 'are', 'and', 'of', 'Right',
'ought', 'to', 'be', 'Free', 'and', 'Independent', 'States', 'that', 'they', 'are', 'Absolved',
'from', 'all', 'Allegiance', 'to', 'the', 'British', 'Crown', 'and', 'that', 'all', 'political',
'connection', 'between', 'them', 'and', 'the', 'State', 'of', 'Great', 'Britain', 'is', 'and',
'ought', 'to', 'be', 'totally', 'dissolved', 'and', 'that', 'as', 'Free', 'and', 'Independent',
'States', 'they', 'have', 'full', 'Power', 'to', 'levy', 'War', 'conclude', 'Peace', 'contract',
'Alliances', 'establish', 'Commerce', 'and', 'to', 'do', 'all', 'other', 'Acts', 'and', 'Things',
'which', 'Independent', 'States', 'may', 'of', 'right', `do']
```

We want to write an algorithms that finds the distance between the closest pair of equal words. In the example, this would be ‘the’ in position 2 and position 5.

(a) The naive algorithm takes all pairs of words in the array, compare them, and if they are equal, calculate their distance. By maintaining the best seen distance, we can thus identify the word repeated with minimum distance. What is the asymptotic runtime of this algorithm.

(b) Use a Linear Hash File in order to create a linear-time algorithm. You can describe the algorithm in words.

Problem 3:

You are working for a large employee-owned company that is facing financial difficulties. The employees have decided to implement a salary cap. All employees below a given cap C will retain their salary, whereas everyone making above C will see their salary temporarily reduced to C. You are given an array of current salaries and a maximum payroll of P.

Create an algorithm of run-time $O(n \log(n))$ to calculate a cap such that the total of adjusted salaries is as close as possible, but smaller than P.

(Hint: use sorting)

Problem 4:

Write a **divide-and-conquer** algorithm that finds the maximum difference between two elements in an array. These two elements would be necessarily the minimum and the maximum. Use the MT to calculate the asymptotic runtime.

Problem 5:

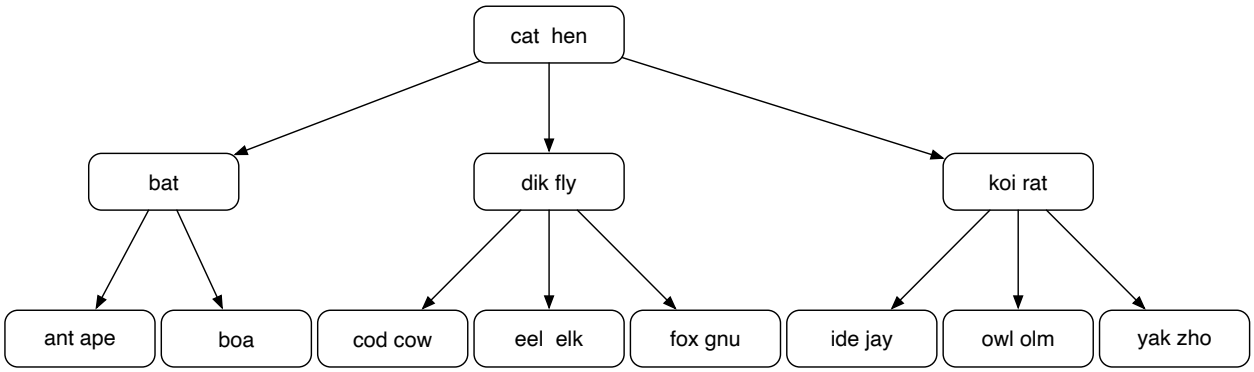
What is the file state of a Linear Hashing File with 9 buckets.

What are the buckets in which we would place records with (hash of) key: 20, 25, 30, 35, and 40.

Problem 6:

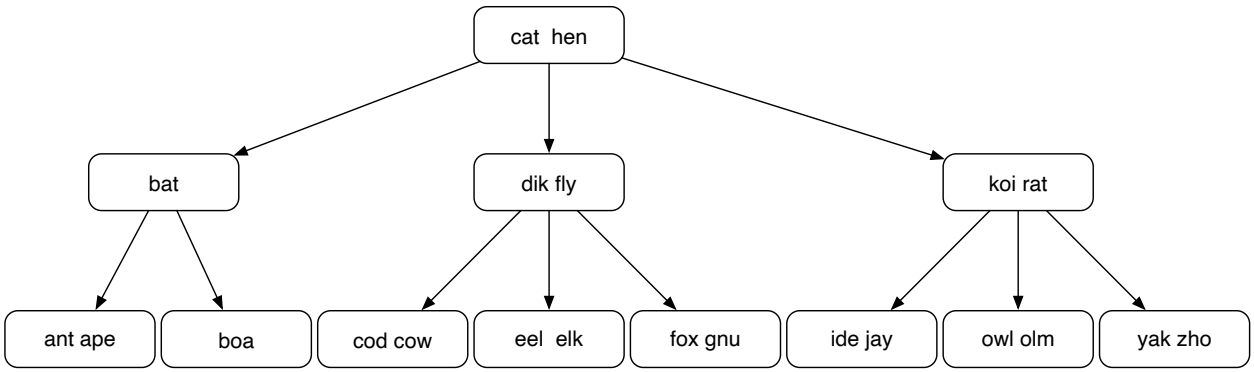
We are given a 2-3 tree in the following state:

(a)



Insert “rok”

(b) Given



insert “emu”.

Show each step.