CS/COE 1501 Midterm Exam 8:30pm

Jay Patel

TOTAL POINTS

25 / 42

QUESTION 1

1 1.a.1 **2 / 2**

✓ **- 0 pts** Correct

- 2 pts Incorrect

QUESTION 2

2 1.a.2 **0 / 2**

- 0 pts Correct

✓ **- 2 pts** Incorrect

- 1 pts Incorrect explanation

QUESTION 3

3 1.a.3 **1 / 2**

- 0 pts Correct

- 2 pts Incorrect

✓ **- 1 pts** Incomplete/incorrect explanation

QUESTION 4

4 1.b.1 **0 / 1**

- 0 pts Multiplicative constants only

✓ **- 1 pts** Wrong Answer

QUESTION 5

5 1.b.2 **0 / 1**

- 0 pts Correct (2^8 = 256)

✓ **- 1 pts** Incorrect

- 0.5 pts Partially correct

QUESTION 6

6 1.b.3.1 **0 / 1**

- 0 pts The letter frequencies are almost equal.

✓ **- 1 pts** Click here to replace this description.

- 0.5 pts Not quite right.

QUESTION 7

7 1.b.3.2 **0 / 1**

- 0 pts Correct

- 1 pts LZW

✓ **- 1 pts** Wrong

- 0.5 pts Not quite right.

QUESTION 8

8 1.b.4.1 **0 / 1**

- 0 pts Theta(# bits in key)

✓ **- 1 pts** Incorrect

QUESTION 9

9 1.b.4.2 **0 / 1**

- 0 pts Theta(# bits in key)

✓ **- 1 pts** Incorrect

QUESTION 10

10 1.c **4 / 4**

✓ **- 0 pts** Correct

- 4 pts Incorrect

- 0.5 pts RST is different from DLB

QUESTION 11

11 2.a.1 **3 / 3**

✓ **- 0 pts** Correct

- 2 pts Incorrect

QUESTION 12

12 2.a.2 **2 / 4**

- 0 pts Correct

- 3 pts Incorrect

✓ **- 2 pts** Partially Correct

- 0.5 pts Small mistake

QUESTION 13

13 2.b.1 **7 / 8**

- 0 pts Correct Tree

- **8 pts** Incorrect
- **3 pts** Wrong bitstring representation

✓ - **1 pts** **Small mistake in bitstring representation**

✓ - **1 pts** **Incorrect**

- **0.5 pts** Small mistake

QUESTION 14

14 2.c.1 **0 / 2**

- **0 pts** Correct

✓ - **2 pts** **Wrong**

- **1 pts** Partially incorrect

QUESTION 15

15 1.c.2.1 **1.5 / 1.5**

✓ - **0 pts** **Correct**

- **1.5 pts** Incorrect

this is closed addressing

QUESTION 16

16 1.c.2.2 **0.5 / 1.5**

- **0 pts** Correct

✓ - **1 pts** **Incorrect/too slow**

QUESTION 17

17 1.c.2.3 **0.5 / 1.5**

- **0 pts** Correct

✓ - **1 pts** **Incorrect/too slow/not clear**

- **0.5 pts** Small mistake

QUESTION 18

18 1.c.2.4 **0.5 / 0.5**

✓ - **0 pts** **Correct**

- **0.5 pts** Need to explicitly state your assumptions

QUESTION 19

19 1.c.3 **2 / 2**

✓ - **0 pts** **Correct**

- **1 pts** Not general enough
- **1.5 pts** Too specific

QUESTION 20

20 2.c.2 **1 / 2**

- **0 pts** Correct
- **2 pts** Incorrect

University of Pittsburgh
CS/COE 1501: Algorithm Implementation
Dr. Sherif Khattab

Midterm Exam
October 11, 2018

Midterm Exam (40 pts)

(Thursday 10/11 20:30-21:45pm)

Instructions:

- You have 75 minutes to solve the 2 questions of the exam.
- Attempt all questions. **The exam will be graded out of 40 points (i.e., there are 2 bonus points).**
- The exam is closed-book and closed-notes.
- Please use *a pen or a dark pencil.*
- Answer each question in the space provided for it. Do not answer a problem (or part of it) in the space for another problem. If you need extra space use additional sheets. Remember to write your name and problem number on any additional sheets you use.
- Plan your time wisely. It is recommended to read all problems through first. Do not spend too much time on any single problem. Make sure that your answers are clear and neat.

Question:	1	2	Total
Points:	23	19	42
Score:			

Name: Jay Patel PeopleSoft ID: 4039189

Question 1

(a) **True/False.** Indicate whether each of the following statements is True or False. For False answers, **explain** why it is false.

1. (2 marks) ☐ By compressing data, lossily or losslessly, we represent the same information in less space.

False, when we use lossily, we would lose some information for ex: in .mp3 file, we would lose bitrate outside of what human can notice difference.

2. (2 marks) ☐ When the LZW compression algorithm detects a codeword that is not inside its codebook, it reports error and halts.

True

3. (2 marks) ☐ It is a good rule of thumb to keep a linear-probing hash table at least 50% full.

False, don't need to be half full.

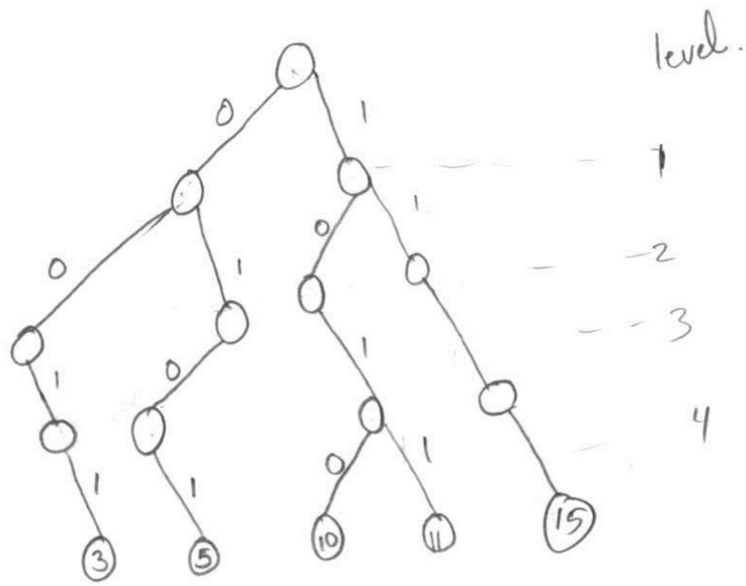
(b) **Fill in the Blanks:** Complete the statements below with the *most appropriate* words or phrases.

1. (1 mark) [greater than equal to] are ignored in Big-O asymptotic analysis but not in Tilda-approximation.
2. (1 mark) The maximum number of children of each node in a multiway trie that considers 8 bits at a time is [2].
3. (2 marks) On the string "AAAAAAAAAABBBBBBBBBBCCCCCCCCCCC", Huffman encoding will obtain no compression because [of string repetition.], whereas [KMP] will be best suited to compress the string.
4. (2 marks) The worst-case runtime to insert into a **Digital Search Tree** is [$\Theta(\log n)$], whereas the average-case runtime to insert into a **Radix Search Trie** is [$\Theta(m)$].

dept of trie

(c) **Short Answer**

1. (4 marks) Draw the result of inserting the following keys (as 4 bit integers) into a Radix Search Trie:
   - 5 (0101)
   - 10 (1010)
   - 11 (1011)
   - 3 (0011)
   - 15 (1111)

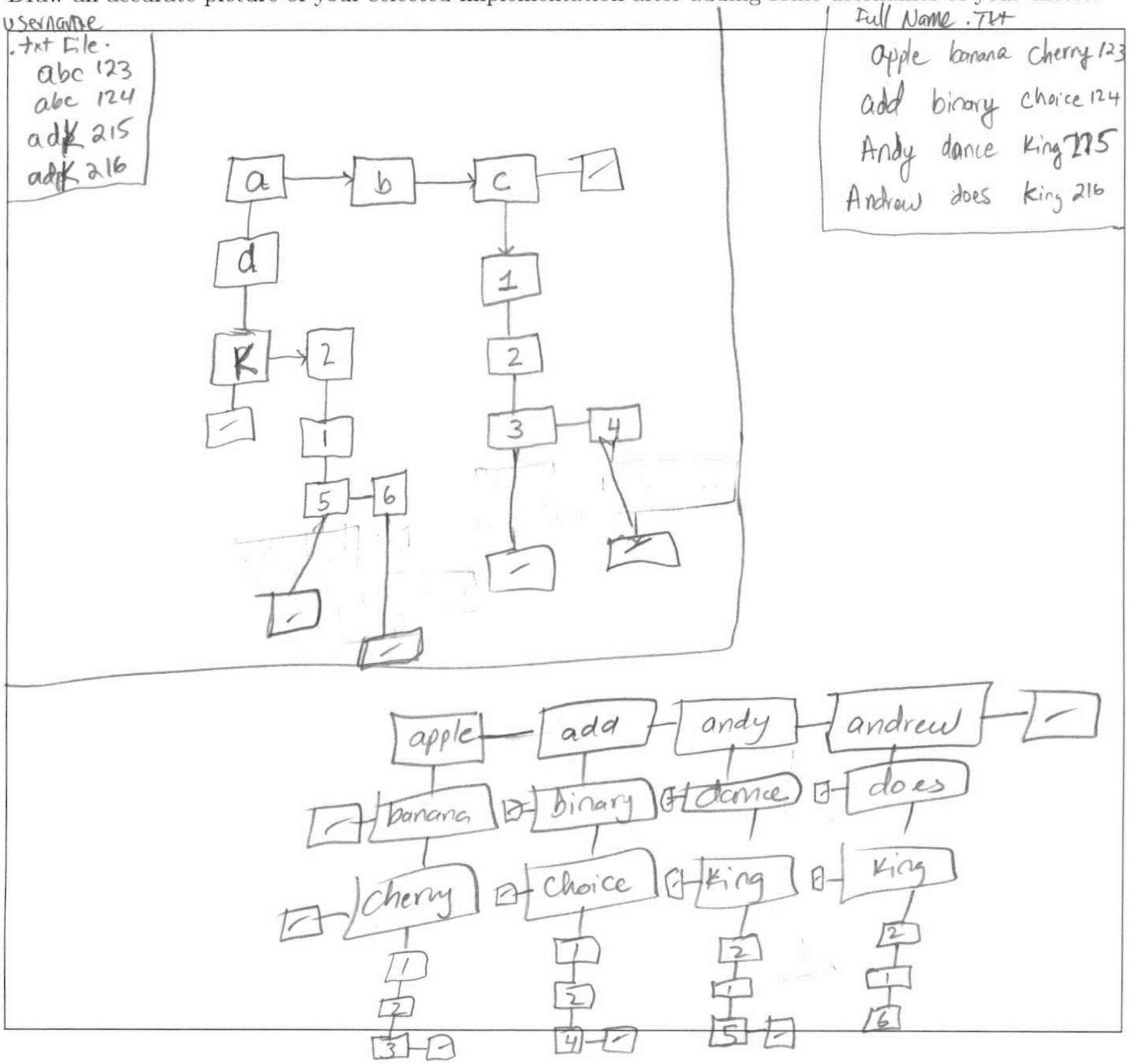


The number of levels of your trie is 4.

2. (5 marks) Assume that you have been tasked with building a symbol table that will map Pitt usernames to full names (e.g., the key "abc123" would map to the value "Bot Anonymous"). Further, assume that you will be using this symbol table to perform the following operations:
   - **Operation 1:** Given a username, return the associated full name.
   - **Operation 2:** Given a sequence of 3 characters (e.g., "abc"), determine whether those initials are currently being used as a username, and if so, what the next available number should be (e.g., "abc123" exists, so 124 is the next available number.

   What symbol table implementation would you use? DLB

   Draw an accurate picture of your selected implementation after adding some usernames of your choice.



Explain how each of the above two operations will be supported by your data structure.

Operation 1:

It fills the trie with username.txt file, and also fill 2nd trie with full name.txt.

If asked for abc 123 → it will return

value at (fullname.txt, 1) → apple banana Cherry 123.

What is the asymptotic worst-case running time of Operation 1 using your data structure?

$O(\log_2 n)$

Operation 2:

Username.txt, will fill DIB – with usernames, if found here; username is used, as for #, it assumes that number before is occupied and after is free.

What is the asymptotic worst-case running time of Operation 2 using your data structure?

$O(n \log n)$

State any assumptions that you make.

for DIB for Username: If number is present, the number before is also present. for ex: if 123, is present, 122 is also present & can't be used.

3. (2 marks) Briefly explain the two main factors that influence the choice of a data structure to store a given set of data items (Hint: Think about how we selected an appropriate data structure to store the Huffman Trie or the LZW codebook, for example).
I will consider:

Run-Time	and
Space/Memory	

Total for Question 1: 23 marks

Question 2

(a) Consider the following two open-addressing hash tables, with $h(x) = x \bmod 13$, shown below. Also, consider the following keys (in order): `17, 24, 30, 37, 13, 18`.

Linear Probing	
Index	Key
0	13
1	
2	
3	
4	17
5	30
6	18
7	
8	
9	
10	
11	24
12	37

collisions
111

Double Hashing	
Index	Key
0	30
1	
2	
3	37
4	17
5	18
6	13
7	
8	
9	
10	
11	24
12	

$(x \bmod 11) + 1$

$h_2(30) = 9$ 111

$h_2(37) = 5$

$h_2(13) = 3$ - (2 collis

1

1. (3 marks) Assume that linear probing is being used for collision resolution. **Show** the table after the keys shown above are inserted in the order shown above.
The **total number of collisions** using linear probing is $\boxed{3}$.

2. (4 marks) Assume now that double hashing is being used for collision resolution, with $h_2(x) = (x \bmod 11) + 1$. **Show** the table after the keys shown above are inserted in the order shown above. For full credit for each collision indicate the $h_2(x)$ value for the key.
The **total number of collisions** using double hashing is $\boxed{4}$.

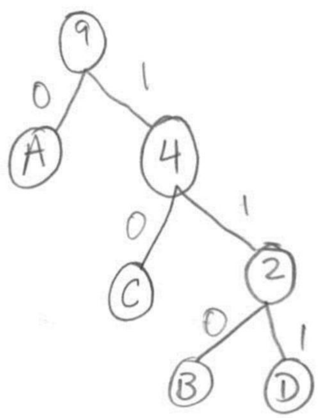
(b) **Huffman Compression.** Consider a file containing the following text data: `AAABCCAAD`.

1. (5 marks) Draw the Huffman trie for the given file. **Show all steps.**

CW

65 A - 5 - 0
66 B - 1 - 110
67 C - 2 - 10
68 D - 1 - 111

128 64 32 16 8 4 2 1
0 1 0 0 0 0 0 1 - A
0 1 0 0 0 0 1 0 - B
0 1 0 0 0 0 1 1 - C
0 1 0 0 0 1 00 - D
0 1



The number of levels of your trie is 3.

The codeword for 'A' is 0.

2. (3 marks) Show the **bitstring representation** of the resulting Huffman trie when it is to be stored inside the compressed file. The ASCII code of 'A' is 65.

0 01000001 1 0 01000011 1 0 01000010 1 01000100

A C B D

(c) (4 marks) Complete the code for the KMP string matching algorithm below, which will return the first index `i` of string `txt` such that each of the `m` characters in the substring of `txt` starting at `i` matches each character in `pat`. Assume the 2-dimensional array `dfa` has been initialized as discussed in class.

```
public int kmp_search(String pat, String txt) {
    int m = pat.length();
    int n = txt.length();
    int i, j;
    for (i = 0, j = 0; i < n && j < m; i++)
        j = S( n[i+1] , m[j] , j+1 );

    if (j == m) return i;

    return n; // not found
}
```

I knew, but forgot during exam. :(

Total for Question 2: 19 marks