

Homework 5

● Graded

Student

Giancarlos Marte

Total Points

99 / 103 pts

Question 1

Huffman's Algorithm

21 / 22 pts

– 0 pts Correct

– 22 pts Not answered

– 7 pts a not answered

– 5 pts b not answered/completely wrong

– 5 pts c not answered

– 5 pts d not answered

– 5 pts wrong trie

– 2 pts wrong coding

✓ – 1 pt d not complete/incorrect

– 2 pts c not complete

Question 2

Fixed-length binary codes

6 / 8 pts

– 0 pts Correct

✓ – 2 pts not minimal/wrong order

– 3 pts not complete/wrong

– 8 pts not answered

Question 3

Adjacency lists for an undirected Graph

5 / 5 pts

✓ – 0 pts Correct

– 1 pt incorrect item

– 5 pts not answered

Question 4

DFS of an undirected Graph

10 / 10 pts

✓ - 0 pts Correct

- 2 pts not complete/ incorrect
- 10 pts not answered/ unreadable
- 4 pts not complete

Question 5

BFS of an undirected Graph

5 / 5 pts

✓ - 0 pts Correct

- 2 pts not complete/incorrect
- 5 pts not answered
- 3 pts not complete/incorrect

Question 6

Adjacency lists for a directed Graph

5 / 5 pts

✓ - 0 pts Correct

- 1 pt incorrect item
- 3 pts incorrect items
- 5 pts not answered

Question 7

App scemario: word letter substitutions

15 / 16 pts

- 0 pts Correct
- 2 pts a not correct
- 2 pts b not correct
- 2 pts d not correct/incomplete

✓ - 1 pt not correct

- 16 pts not answered

Question 8

App scenario: prerequisites

16 / 16 pts

✓ - 0 pts Correct

- 2 pts b not correct
- 16 pts not answered
- 4 pts d not answered
- 4 pts a not answered

Question 9

App scenario: Kevin Bacon game

16 / 16 pts

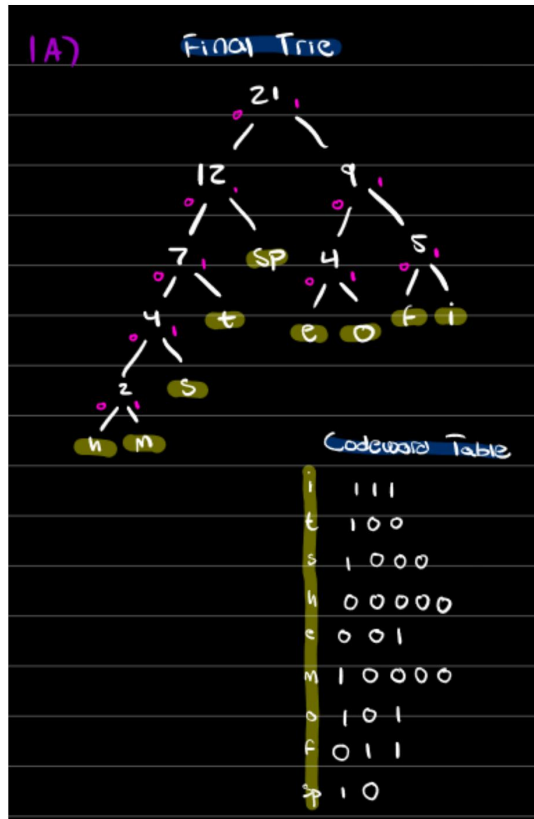
✓ - 0 pts Correct

- 16 pts not answered
- 2 pts b not correct
- 4 pts b not answered
- 4 pts c not answered
- 4 pts d not answered

Question assigned to the following page: [1](#)

Giancarlos Marte
hw5 (cs310)
5/4/21

1. Huffman's Algorithm



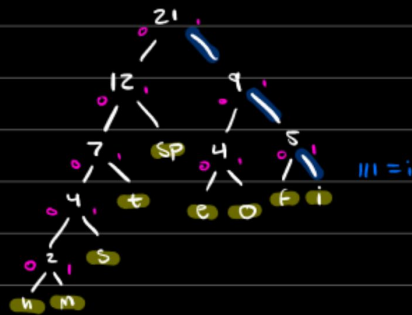
1B)
111 100 10 111 1000 10 100 00000 001 10 100 111 10000 001 10 101 011 10 011 101 101 1000
i t sp i s sp t h e sp t i m e sp o f sp f o o s

Question assigned to the following page: [1](#)

1c)

In order to decode you have to go through the tree, following the 1's & 0's (from the compressed bitstring) until you get to the character

111 100 10...



1d)

h	m	s	e	o	f	i	t	sp
1	1	2	2	2	2	3	3	5

h 00000 least freq

sp 10 most freq

Question assigned to the following page: [2](#)

2. Fixed-length binary codes

2)

it is the time of foos

22 char = n

$$n = 2^k \quad k = \text{bits}$$

$$22 = 2^k \rightarrow k = \lceil \log_2 n \rceil$$

$$= \lceil \log_2 22 \rceil$$

5 bit encoding = 5

sp	00000
e	00001
f	00010
h	00011
i	00100
M	00101
O	00111
S	01000
t	01001

Question assigned to the following page: [3](#)

3. Adjacency lists for an undirected graph

3)

[0] → 2 → 5 → 6

[1] → 4 → 8 → 11

[2] → 0 → 3 → 5 → 6

[3] → 2 → 6 → 10

[4] → 1 → 8

[5] → 0 → 2 → 10

[6] → 0 → 2 → 3

[7] → 8 → 11

[8] → 1 → 4 → 7 → 11

[9]

[10] → 3 → 5

[11] → 1 → 7 → 8

Question assigned to the following page: [4](#)

4. DFS of an undirected Graph

4)

dfs(0)

marked[]

adj[]

0	T
1	
2	
3	
4	
5	

0	2 5
1	4
2	0 3 5
3	2
4	1
5	0 2

dfs(2)

marked[]

adj[]

0	T
1	
2	T
3	
4	
5	

0	2 5
1	4
2	0 3 5
3	2
4	1
5	0 2

dfs(3)

DONE

marked[]

adj[]

0	T
1	
2	T
3	T
4	
5	

0	2 5
1	4
2	0 3 5
3	2
4	1
5	0 2

(from 2)

dfs(5)

DONE

marked[]

adj[]

0	T
1	
2	T
3	T
4	
5	T

0	2 5
1	4
2	0 3 5
3	2
4	1
5	0 2

check 0
check 2

Question assigned to the following page: [4](#)

back to 0		DONE	
marked[]		adj[]	
0	T	0	2 5
1		1	4
2	T	2	0 3 5
3	T	3	2
4		4	1
5	T	5	0 2
check 5			

dfs(1)			
marked[]		adj[]	
0	T	0	2 5
1	T	1	4
2	T	2	0 3 5
3	T	3	2
4		4	1
5	T	5	0 2

dfs(4)		DONE	
marked[]		adj[]	
0	T	0	2 5
1	T	1	4
2	T	2	0 3 5
3	T	3	2
4	T	4	1
5	T	5	0 2
check 1			

edgeTo[]	
0	
1	
2	0
3	2
4	1
5	2

Question assigned to the following page: [5](#)

5. BFS of an undirected Graph

5)

queue	marked[]	edgeTo[]	adj[]
0	0 T	0	0 2 5
	1	1	1 4
	2	2	2 0 3 5
	3	3	3 2
	4	4	4 1
	5	5	5 0 2

queue	marked[]	edgeTo[]	adj[]
2	0 T	0	0 2 5
5	1	1	1 4
	2 T	2 0	2 0 3 5
	3	3	3 2
	4	4	4 1
	5 T	5 0	5 0 2

queue	marked[]	edgeTo[]	adj[]
5	0 T	0	0 2 5
3	1	1	1 4
	2 T	2 0	2 0 3 5
	3 T	3 2	3 2
	4	4	4 1
	5 T	5 0	5 0 2

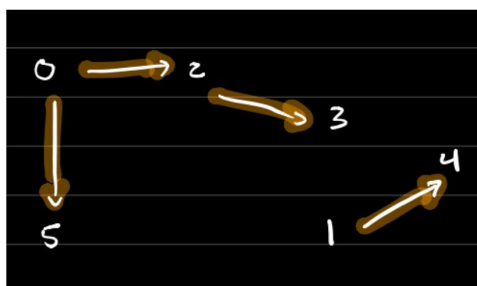
queue	marked[]	edgeTo[]	adj[]
3	0 T	0	0 2 5
	1	1	1 4
	2 T	2 0	2 0 3 5
	3 T	3 2	3 2
	4	4	4 1
	5 T	5 0	5 0 2

Question assigned to the following page: [5](#)

queue	marked[]	edgeTo[]	adj[]
0	T	0	2 5
1		1	4
2	T	2 0	0 3 5
3	T	3 2	2
4		4	1
5	T	5 0	0 2

queue	marked[]	edgeTo[]	adj[]		
1	0	T	0	2 5	
1	T	1	1	4	
2	T	2	0	2	0 3 5
3	T	3	2	3	2
4		4	4	4	1
5	T	5	0	5	0 2

queue	marked[]	edgeTo[]	adj[]		
4	0	T	0	2 5	
1	T	1	1	4	
2	T	2	0	2	0 3 5
3	T	3	2	3	2
4	T	4	1	4	1
5	T	5	0	5	0 2



Question assigned to the following page: [6](#)

6. Adjacency lists for a directed graph

6)

[0] → 5 → 6

[1]

[2] → 0 → 3

[3] → 6 → 10

[4] → 1

[5] → 2 → 10

[6] → 2

[7] → 8 → 11

[8] → 1 → 4

[9]

[10] → 3

[11] → 8

Question assigned to the following page: [Z](#)

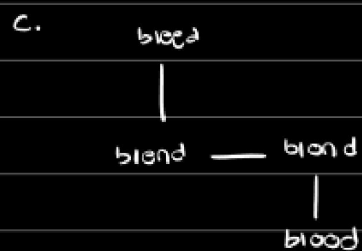
7, 8 & 9. Graphs from application scenarios

7 (b)

a. The vertex is a word.

Examples: bleed, blend, blond, blood

b. An edge is undirected & is formed if a word differs from another word by 1 character.



d. [bleed] → blend
[blend] → blond
[blond] → blood
[blood]

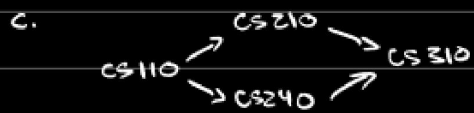
Question assigned to the following page: [8](#)

$B(C)$

a. The vertex is a course.

Examples: CS110, CS210, CS310, CS240

b. An edge is directed & exists if one class is a prerequisite of another.



d.

$[CS110] \rightarrow CS210 \rightarrow CS240$

$[CS210] \rightarrow CS310$

$[CS240] \rightarrow CS310$

$[CS310]$

Question assigned to the following page: [9](#)

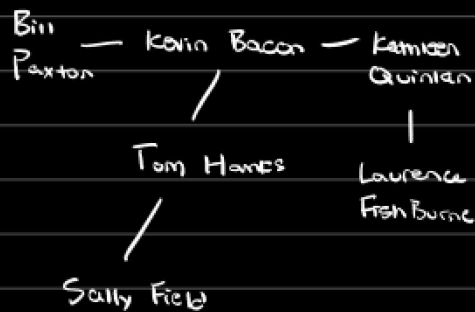
9(D)

a. A vertex is an actor.

Examples: Tom Hanks, Sally Field

b. An edge is undirected and exists if an actor has shared a movie with another actor that has shared a movie with Kevin Bacon or another actor who has.

c.



d.

[Kevin Bacon] → Tom H. → Kathleen Q.
↳ Bill P.

[Tom Hanks] → Sally F.

[Sally Field]

[Lawrence
Fishburne]

[Bill Paxton]

[Kathleen
Quinlan] → Lawrence F.