COMPS265F

Take-home Assignment

25. May. Tue

Name: Jiawei Wang

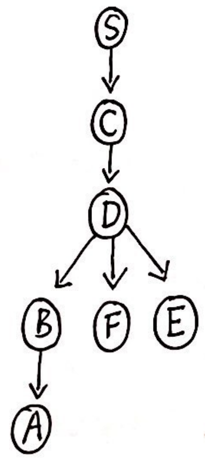
ID: S1239587

Question 1 (10 marks)

(a) (6)

discovered order	1	2	3	4	5	6	7
vertex v	S	C	D	B	A	F	E
d[v]	1	2	3	4	5	8	10
f[v]	14	13	12	7	6	9	11
π[v]	–	S	C	D	B	D	D

Below is the depth-first tree obtained:



(b) (2)

edge	(S,C)	(C,A)	(C,D)	(D,B)	(B,A)	(D,E)	(D,F)	(E,C)
type	tree	forward	tree	tree	tree	tree	tree	back

(C) (2)

The above directed graph do not have a topological sort.
Since the DFS tree obtained has back edge which means the graph has some cycles.
Then it is no a directed acyclic graphs.

Question 2 (15 marks)

(a) (5)

For the NFA, the transition table f_ε with the lambda closures is:

f_ε	s	a	b	c	ε	$\lambda(s)$
start	0	$\{1\}$	ϕ	ϕ	ϕ	$\{0\}$
	1	ϕ	ϕ	ϕ	$\{2\}$	$\{1,2\}$
	2	$\{2,3\}$	ϕ	ϕ	$\{4\}$	$\{2,3,4\}$
	3	ϕ	ϕ	$\{2\}$	ϕ	$\{2,3,4\}$
	4	ϕ	$\{5\}$	ϕ	ϕ	$\{4,5\}$
final	5	ϕ	ϕ	ϕ	ϕ	$\{5\}$

(b) (10)

Let f_D be the transition function of the DFA.

Start state: $=\lambda(0) = \{0\}$ $\quad f_D(\{0\},a) = \lambda(f_\varepsilon(0,a)) = \lambda(1) = \{1,2\}$

$f_D(\{1,2\},a) = \lambda(f_\varepsilon(1,a) \cup f_\varepsilon(2,a)) = \lambda(\{2,3\}) = \{2,3,4\}$.

$f_D(\{2,3,4\},a) = \lambda(f_\varepsilon(2,a) \cup f_\varepsilon(3,a) \cup f_\varepsilon(4,a)) = \lambda(\{2,3\}) = \{2,3,4\}$.

$f_D(\{2,3,4\},b) = \lambda(f_\varepsilon(2,b) \cup f_\varepsilon(3,b) \cup f_\varepsilon(4,b)) = \lambda(5) = \{5\}$

$f_D(\{2,3,4\},c) = \lambda(f_\varepsilon(2,c) \cup f_\varepsilon(3,c) \cup f_\varepsilon(4,c)) = \lambda(2) = \{2,3,4\}$.

$f_D(\{5\},a) = f_D(\{5\},b) = f_D(\{5\},c) = \phi$

Therefore, we have the following table:

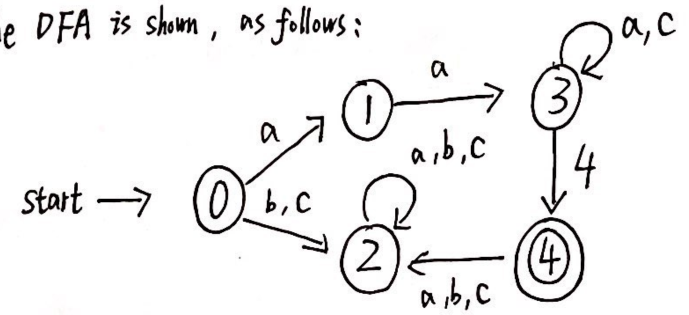
	f_D	a	b	c
start	$\{0\}$	$\{1,2\}$	ϕ	ϕ
	$\{1,2\}$	$\{2,3,4\}$	ϕ	ϕ
	$\{2,3,4\}$	$\{2,3,4\}$	$\{5\}$	$\{2,3,4\}$
final	$\{5\}$	ϕ	ϕ	ϕ
	ϕ	ϕ	ϕ	ϕ

Question 2 (b) cont'd:

Renaming the table:

fp	S	a	b	c
start	0	1	2	2
	1	3	2	2
	3	3	4	3
final	4	2	2	2
	2	2	2	2

The DFA is shown, as follows:



Question 3 (5 marks)

Suppose, for the sake of contradiction, that L is regular. Thus, L can be accepted by a DFA with m states with m state for some $m \in N$.

Consider the string $a^m b^m c^m$, since $|a^m b^m c^m| \geq m$, by the pumping lemma, there are string x, y, z, such that:

① $|y| > 0$

② $a^m b^m c^m = xyz$

③ $|xy| \leq m$.

④ $xy^iz \in L$ for any $i \in N$

then $a^m b^m c^m$ cannot contain a and c at the same time.

And we can pump y to y^i $(i > 1)$, for sufficiently large i.

the $xy^iz \notin L$, which is a contradiction.

Thus, L is not regular

Question 4 (15 marks)

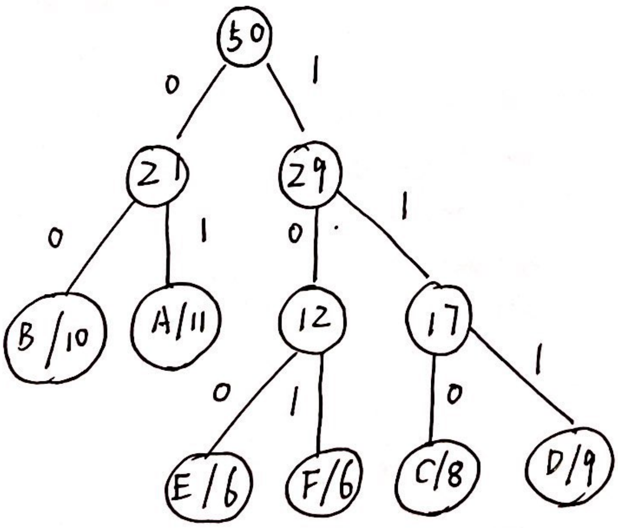
(a) (5)

The following node tree do not have the minimum average character length.
since we need to find the smallest first and merge them.
For example: in the given graph. "E/6" and "F/6" they are the smallest two.
we should merge them into a group.

(b) (10)

Steps to construct the Huffman code tree:

1. Merge E & F to (E, F)
2. Merge C & D to (C, D).
3. Merge A & B to (A, B)
4. Merge (E,F) & (C,D) to (E, F, C, D)
5. Merge (E,F,C,D) & (A,B) to (A, B, E, F, C, D)



Char	Huffman code
A	01
B	00
C	110
D	111
E	100
F	101

The average character length:

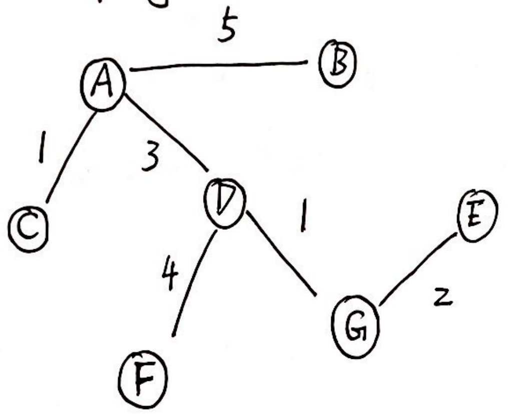
$$L_C(S) = \frac{3\times6+3\times6+3\times8+3\times9+2\times10+2\times11}{50} = \frac{129}{50} = 2.58$$

Question 5 (20 marks)

(a) (5)

order	1	2	3	4	5	6	7	8	9	10
edge	A,C	D,G	E,G	A,D	F,D	A,B	B,E	F,G	F,C	B,D
weight	1	1	2	3	4	5	5	5	5	6
include or not	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No

The minimum spanning tree:



The weight (minimum)
$= 1+1+2+3+4+5$
$= 16$

(b) (10)

From the table in (a), we can get the sequence

order add $1^{\#}$: $\rightarrow$ (A, C) (D, C), since both of them had the lowest costs

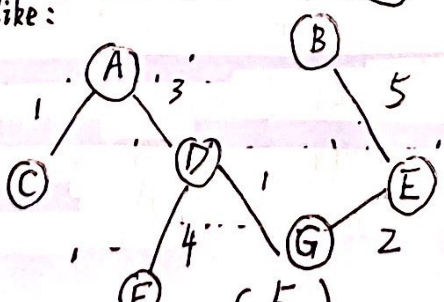
order add $2^{\#}$ $\rightarrow$ (E, G), The edge has the lowest weight in the remaining edges.

order add $3^{\#}$ $\begin{cases} (A, D) \\ (F, D) \end{cases}$, ① The edge has the lowest weight in the remaining edges
② Do not have cycle with current mst edgs.

order add $4^{\#}$ $\rightarrow$ (A, B). In the remaining edgs, (A, B) is the only one with minimum weight and do not have cycle after adding it into current MST.

(c) (5)

Yes, the graph has more than one minimum spanning tree.
The other tree is like:



(5)

Scanned with CamScanner

Question 6 (15 marks)

(a) (2) the output is 1.

(b) (3) This function is trying to find the minimum interval of all adjacent elements in the input list.

(c) (5) Proof:

Analysis: Because of the min( ) function in the return statement. We can conclude that:

"If this program compare all adjacent intervals of the List, then call min( ) function"

We can prove that this program can achieve our goal.

Base case: When L only have 2 elements, then return the difference between them.

Inductive step: when L have n elements $(n > 2)$ $a_0 \sim a_{n-1}$

We compare $[0, m)$ $[m-1, m]$ $[m, n-1]$. which will make comparisions between all adjcent elements in the list.

(d) (5)

From the function, we can know that:

$$func(L) = 2 \cdot func(\frac{L}{2}) + O(1)$$

min( ) of 3 elements take $O(1)$ time

if takes $O(1)$ time

caculate m takes $O(1)$ time.

We use the substitution method to get the solution:

① we guess the solution is $func(n) = O(n)$

② $func(n) \le c \cdot \frac{n}{2} + c \cdot \frac{n}{2} + 1 = cn + 1$. (x) (when $func(n) \le cn$.)

new guess: $func(n) \le cn - d$, now we have:

$$\therefore func(n) \le c((\frac{n}{2}) - d) + c((\frac{n}{2}) - d) + 1 = cn - 2d + 1 \le cn - d.$$

proved.

So the time complexity of $func(n)$ is $O(n)$

(6)

Question 7 (20 marks)

(a) (15)

Please check 97.py

(b) (5)

Since $1 \leq m \leq n$. then in function subseq (s, t).

We only loop the t <n elements>. and Let the value of key value plus one in the defaultdict.

Which cost $O(n)$ time.

The loop s. which has less than n elements.

The total time complexity $T(n) \leq O(n) + O(n) = O(n)$.

——— End of ASM ———