Jin Cheng.
jt 4434.

version B.

1. True or False.

(a). T

(b) F

(c) F

(d) T

(e) T

(f) F

(g) F

(h) T

(i) T

(j) F

2. which of the following ---- DFS are true?

c, d, b. DFS still can find shortest path But time complexity extreme.

3. Which of the following ---- $O(|V|+|E|)$?

d

(a) First ...

4. Which of the following algo .... efficient implementa

D. Floyd..

5. In --- selection problem...

~~...~~ a. b.

6. Which of the following -- shortest path are true?

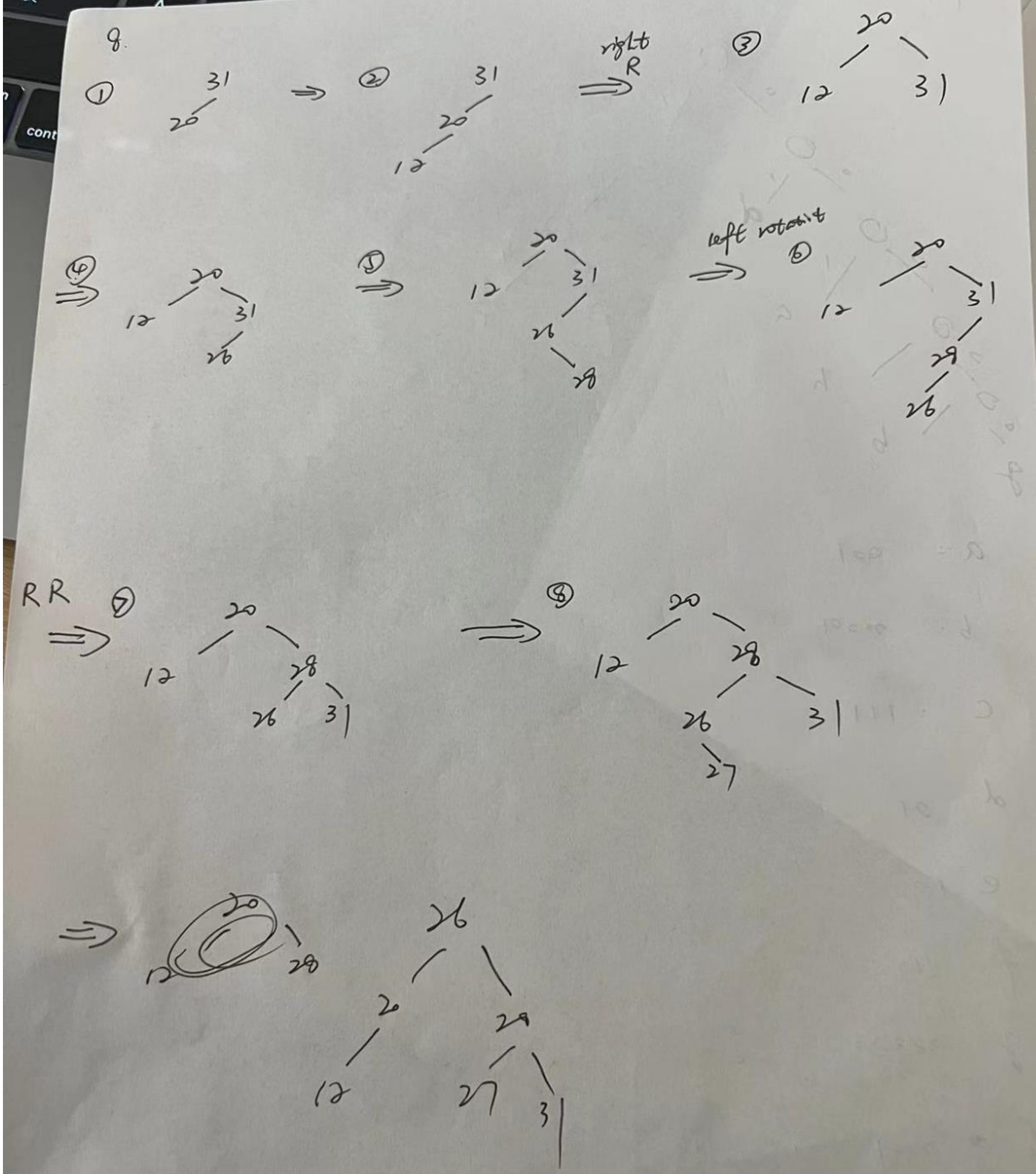
a

7. Which of the following -- are true?

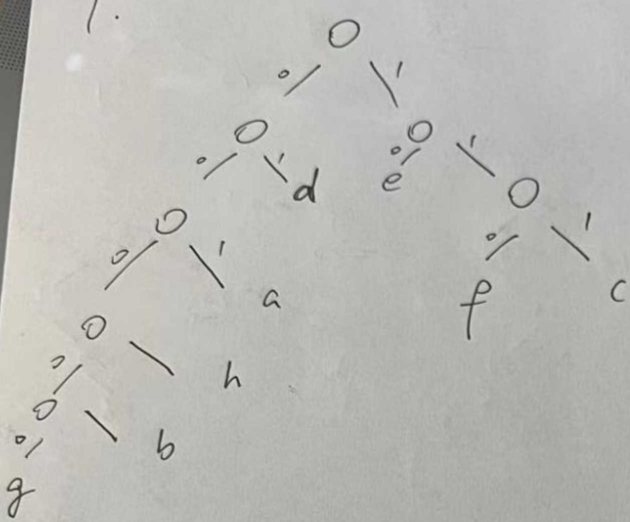
~~...~~ b

~~8.~~

9.



9.



a : 001

b : 00001

c : 111

d 01

e 10

f 110

g 00000

h : 0001

10.

(a) First Five:

A ⇒ E ⇒ G ⇒ C ⇒ B → F

⇒ (A,E) ⇒ (E,G) ⇒ (G,C) ⇒ (C,B), (E,F)

(b)

Kruskal started with the Vertice with lowest weight.

. first. sort the weights.

∴ we have.

weight	source	Dest.
1	A	E
2	H	I
3	E	G
4	G	C
5.	A	G. ⇒ when adding. 5. there is a cycle skip.
6.	C	B
⋮		
15	C	F ⇒ we will reject adding if circle exists.
16	D	G

∴ ~~the order will be~~.

~~(A,E) ⇒ (H,I) ⇒ (E,G) ⇒ (G,C) ⇒ (C,B)~~

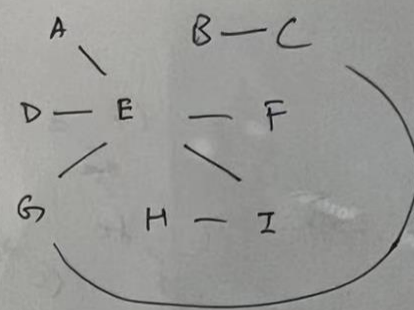
∴ last five are.

(G,C), (B, C) (E,F). (E,I) (D,E)

⇒ (E,G) ⇒ (G,C) ⇒ (C,B), (E,F)

10.b

1	A	E	✓
2	H	I	✓
3	E	G	✓
4	G	C	✓
5	A	G	✗
6	B	C	✓
7	C	E	✗
8	E	F	✓
9	E	I	✓
10	D	E	✓
11	G	H	✗
12	A	B	✗
13	B	E	✗
14	F	I	✗
15	C	F	✗
16	D	G.	✗



11. (a)

Function. Algorithm.

m ← length of B

n ← length of A

DP ← two - Dimenthal table with size (m+1).*(n+1)

with initial value of zeros.

For i = 1, 2, ... n+1

} ⇒ iterate through the two list.

For j = 1, 2, ... , m+1

If A[i-1] == B[j-1] :

DP[i][j] = DP[i-1][j-1] + B[j-1]

if equal, adding to the DP memo table

Else

A[i-1] or B[j-1] will have at least one that not in the lcs

DP[i][j] = FIND-MAX(DP[i-1][j], DP[i][j-1

Return DP[m][n].

(b) The final Result will be

lcs = { 2, 3, 4, 6, 5. }

12.

(a). To prove that greedy is optimal for only nickle and pennies case, we. will only be able to exchange for 4 pennis. since we will be able to use nickels until the total amount is smaller than 5, therefore greedy provides the solution to reduce the chance of using small value if we still can use bigger one.

(b). for the case of dimes, nickles and pennies. as our greedy algorithm, we will keep exchanging the biggest value first until'l we cannot use it, which is dimes in this case. Therefore, I can only use at most 1 nickle here, since if the value is bigger than ¢10, I can use dime. as many as until the reminder is smaller than one dime.

(c). for the case where we can use quarter. dime. nickles, and pennies. as mentioned above, I can at most 2 dimes. otherwise I will be able to use quarter. ~~for example. the value of~~ if the reminder is above ¢30, I will use quarter and nickles as many as I have. Then we will from big to smaller case. . .

13.

(a)

① First, we create a stack to track the vertex and DFS traverse the graph. add vertex into stack.

② Since this is a directed graph, we need to reverse the direction result after our DFS traverse.

③ while stack is not empty.

call DFS for each vertex from stack as the starting vertex.

and