

CSCE 500 Midterm Exam #2

11/14/2018

1. The recurrence of Procedure CUT-ROD(p, n) is given by $T(n) = 1 + \sum_{j=0}^{n-1} T(j)$, with $T(0) = 1$. Solve $T(n)$. (12%)
2. The problem of optimal parenthesization over a chain of matrix multiplications can be solved by a divide-and-conquer approach recursively. Let $m[i, j]$ denote the minimum number of scalar multiplications needed to compute $A_i \bullet A_{i+1} \bullet A_{i+2} \bullet \dots \bullet A_j$, with A_k sized as $p_{k-1} \times p_k$ (for $i \leq k \leq j$), give the recurrence definition of the problem. (10%)

The recurrence leads to exponential complexity but can be solved by dynamic programming much faster. What is the resulting time complexity and how do you get that complexity result? (6%)

3. Given a set of 4 keys, with the following probabilities, determine the cost and the structure of an optimal binary search tree, following the tabular, bottom-up method realized in the procedure of OPTIMAL-BST below to construct and fill $e[1..5, 0..4]$, $w[1..5, 0..4]$, and $root[1..4, 1..4]$.

i	0	1	2	3	4
p_i		0.12	0.08	0.15	0.18
q_i	0.08	0.06	0.09	0.10	0.14

- (a) Fill in the missing 3 statements in the procedure. (6%)
- (b) Construct and fill the three tables, and show the optimal BST obtained. (30%)

OPTIMAL-BST(p, q, n)

```

1  let  $e[1..n+1, 0..n]$ ,  $w[1..n+1, 0..n]$ ,
   and  $root[1..n, 1..n]$  be new tables
2  for  $i = 1$  to  $n+1$ 
3     $e[i, i-1] = q_{i-1}$ 
4     $w[i, i-1] = q_{i-1}$ 
5  for  $l = 1$  to  $n$ 
6    for  $i = 1$  to  $n-l+1$ 
7       $j = i+l-1$ 
8       $e[i, j] = \infty$ 
9       $w[i, j] = w[i, j-1] + p_j + q_j$ 
10     for  $r = i$  to  $j$ 
11       1.  
12       if  $t < e[i, j]$ 
13         2.  
14         3.  
15     return  $e$  and  $root$ 

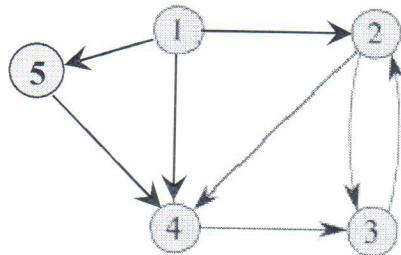
```

4 4

1 2

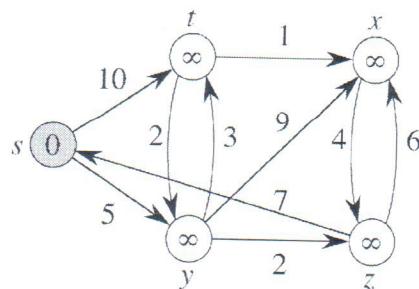
4. Show your construction of an optimal Huffman code for the set of 7 frequencies: **a:2** **b:3**
c:5 **d:8** **e:13** **f:21** **g:34** **h:15** **i:27** **j:9**. (10%)

5. Follow depth-first search (DFS), starting from Node 3, to traverse the graph shown below. Mark (1) the type of every edge and (2) the discovery and the finish times of each node. (10%)



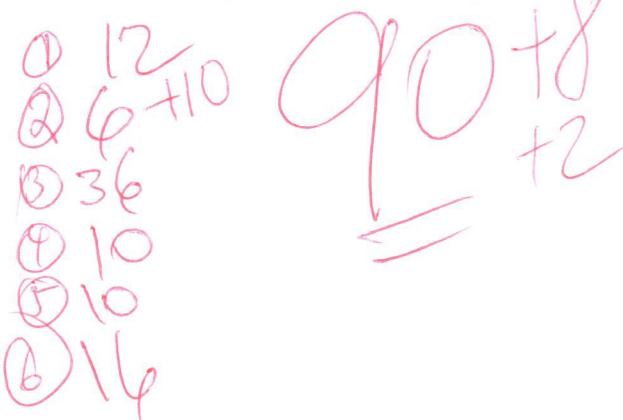
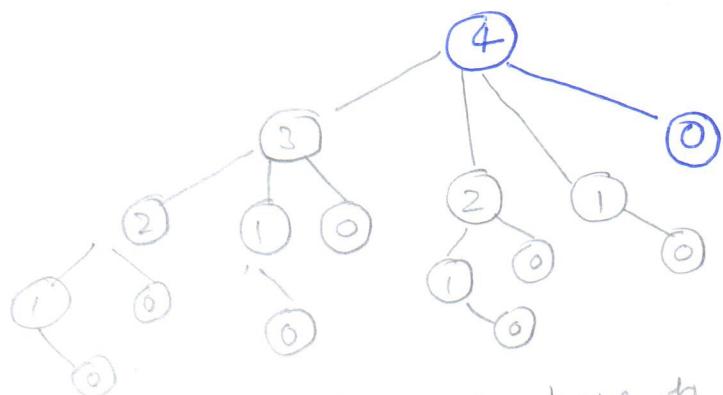
6. The Dijkstra's algorithm (DIJ) solves the single-source shortest-path problem in a weighted directed graph $G = (V, E)$. Given the graph G below, follow DIJ to find shortest paths from vertex s to all other vertexes, with all predecessor edges shaded and estimated distance values from s to all vertexes provided at the end of each iteration. (12%)

What is the time complexity of DIJ for a general graph $G = (V, E)$, if candidate vertexes are kept in a binary min-heap? (4%)



Good luck!

① if $n=4$



for any subproblem we have for 0^{th} node 1 subproblem
 1st node 2 subproblems
 i^{th} node 2^{i-1} subproblems

$$T(0) = 1$$

$$T(n) = 1 + \sum_{j=0}^{n-1} T(j)$$

$$= 1 + \sum_{j=0}^{n-1} 2^{j-1}$$

we know that geometric series $s(n) = \frac{a(r^n - 1)}{r - 1}$

$$T(n) = 1 + 1 \cdot \frac{(2^n - 1)}{2 - 1}$$

$$= 2^n - 1 + 1 = 2^n$$

$\therefore T(n) = 2^n$ OK

$$② m[i, j] = \begin{cases} 0 & i=j \\ \min_{i \leq k < j} m[i, k] + m[k+1, j] + p_{i-1} p_k p_j & i < j \end{cases}$$

$$\therefore T(n) = 1 + \sum_{k=1}^{n-1} T(k) + T(n-k) + 1$$

$$\Rightarrow 2 \sum_{j=0}^{n-2} T(j) + 1$$

$$\geq 2(2^{n-1} - 1) + 1$$

$$\geq 2^n$$

$\therefore T(n) = O(n!) = 2^n$ exponential complexity

for dynamic programming

we will get complexity

$$\begin{aligned} \sum_{l=2}^n \sum_{i=1}^{n-l+1} \sum_{j=i}^l c_{i,j} &= \sum_{l=2}^n \sum_{i=1}^{n-l+1} \sum_{k=i}^{i+l-1} c_{i,k} \\ &= \sum_{l=2}^n \sum_{i=1}^{n-l+1} (l-1) \\ &= \sum_{l=2}^n (l-1)(n-l) \\ &= \sum_{l=2}^n l^2 + \dots \\ &= \frac{1}{6} n(2n+1)(n+1) \\ &= \Theta(n^3) \end{aligned}$$

~~(-8) + 8~~

i.e. Time complexity for dynamic programming reduced to

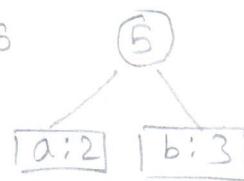
$\Theta(n^3)$

~~(-2) + 2~~

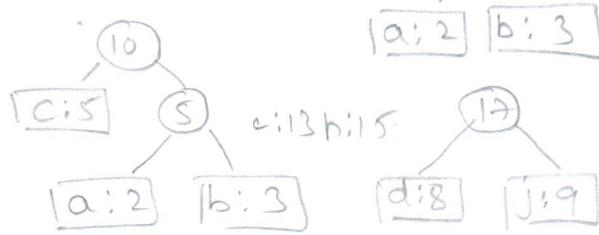
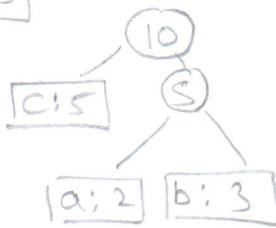
The reason for decreasing time complexity is that it memoizes the data in which there is no need to calculate the subproblems again and again to get the result

④ a:2 b:3 c:5 d:8 j:9 e:13 f:15 g:21 h:27 i:22
 Extract min g:34

c:5 6 d:8 j:9 e:13 h:15 f:21 i:27 g:34

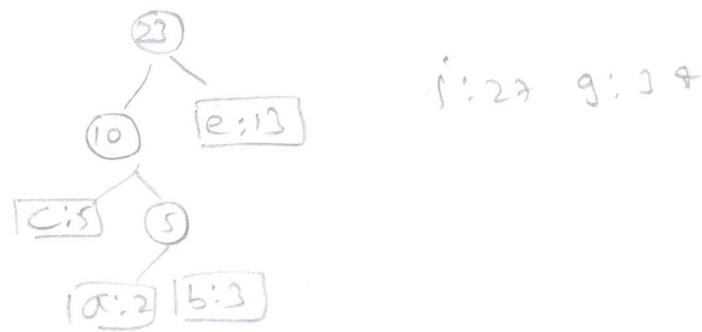
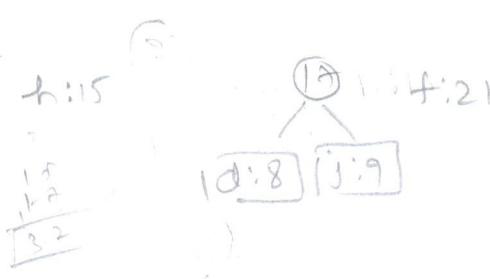


d:8 j:9 10(c:5) s

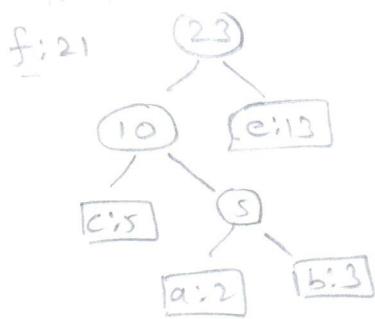


e:13 h:15 f:21 i:27 g:34

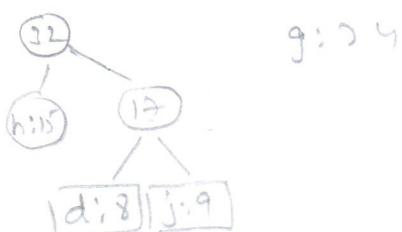
e:13 h:15 f:21 i:27 g:34



i:27 g:34



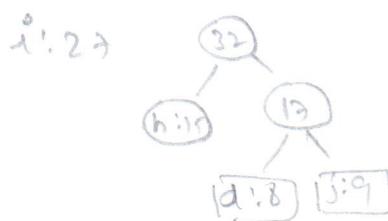
g:27 g:34



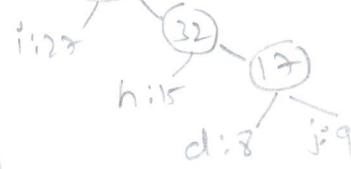
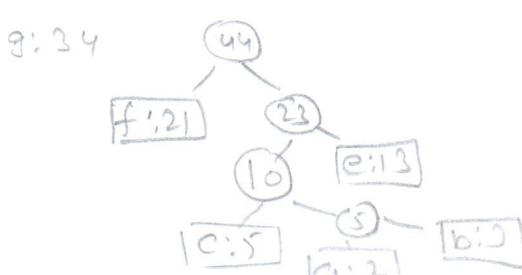
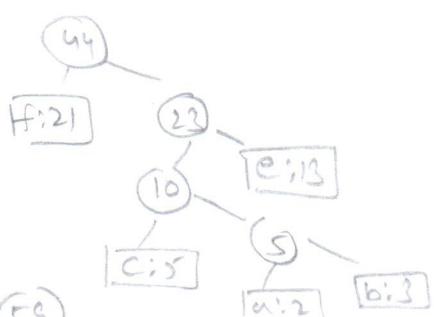
21
22
44

27
32
59

24
47



g:34



g:34

f:21

23

10

s

b:3

a:2

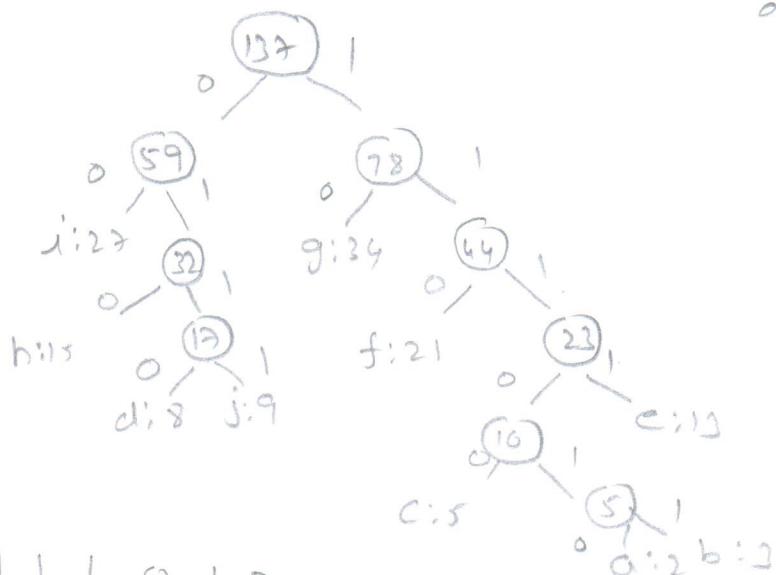
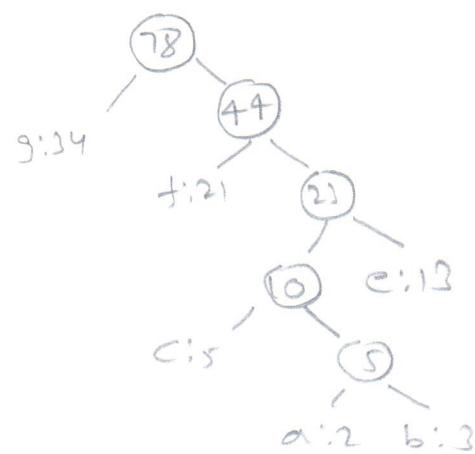
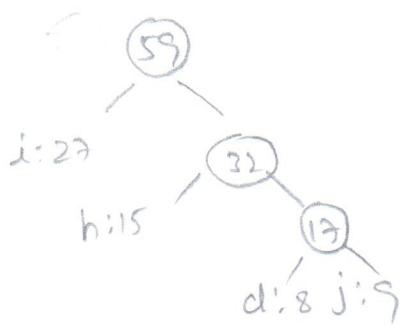
c:5

m:2

h:15

d:8

j:9



a:2 1 1 1 0 1 0

b:3 1 1 1 0 1 1

c:5 1 1 1 0 0

d:8 0 1 1 0

e:13 1 1 1 1

~~OK~~

f:21 1 1 0

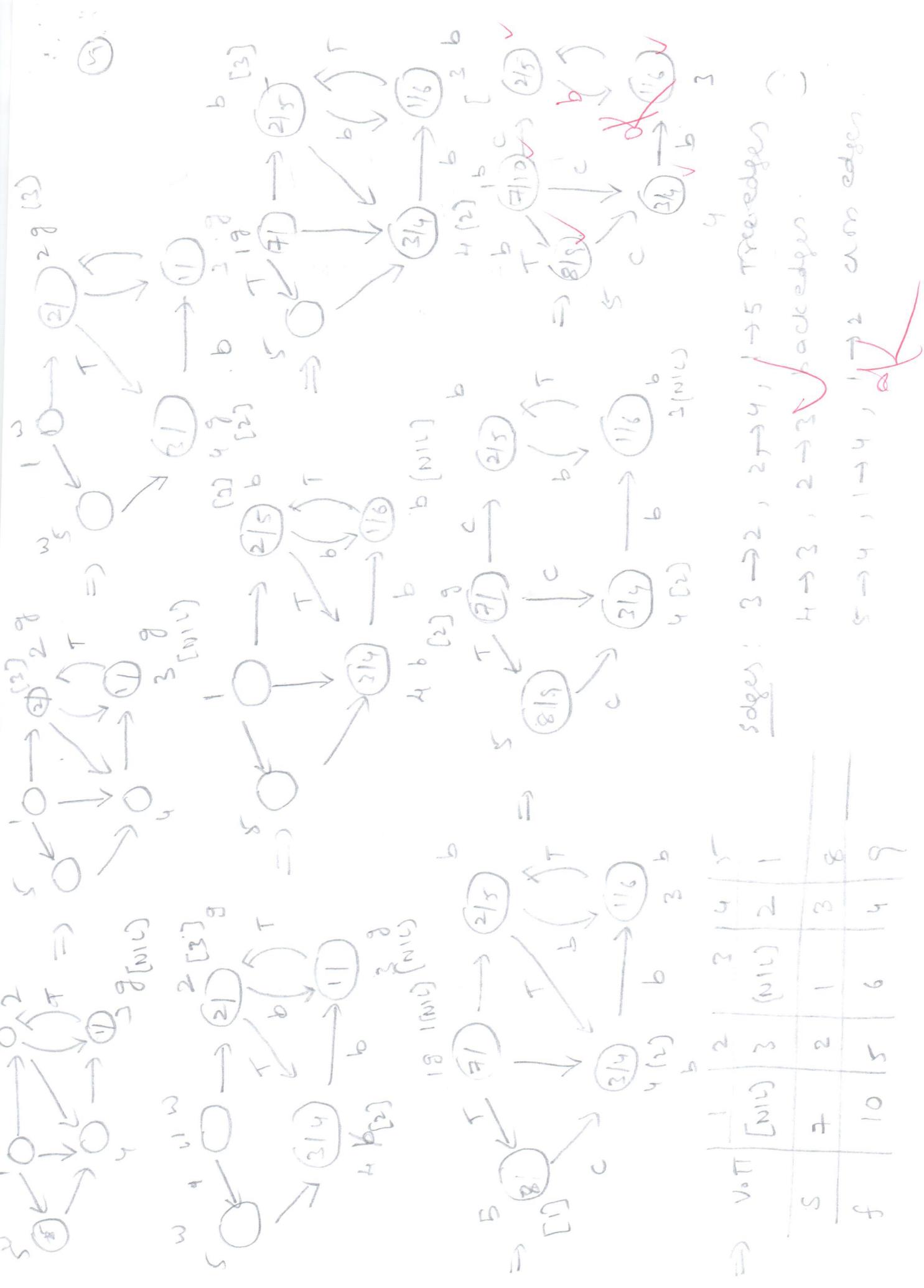
g:34 1 0

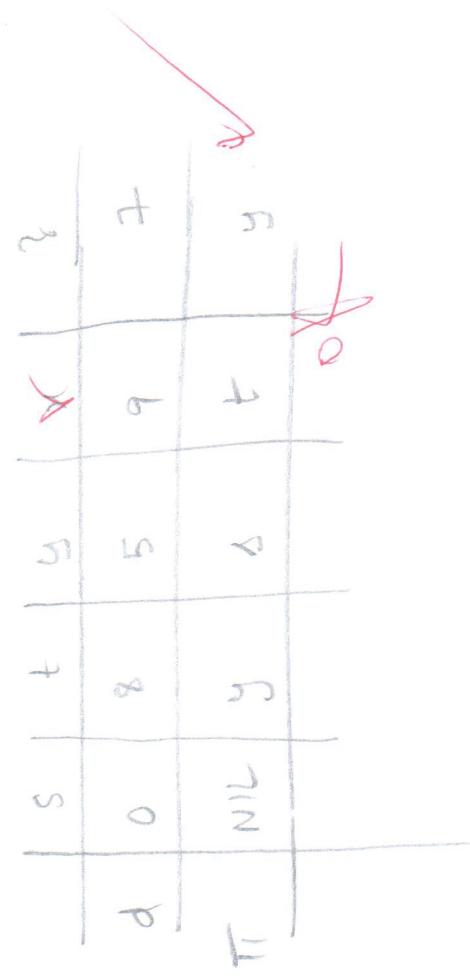
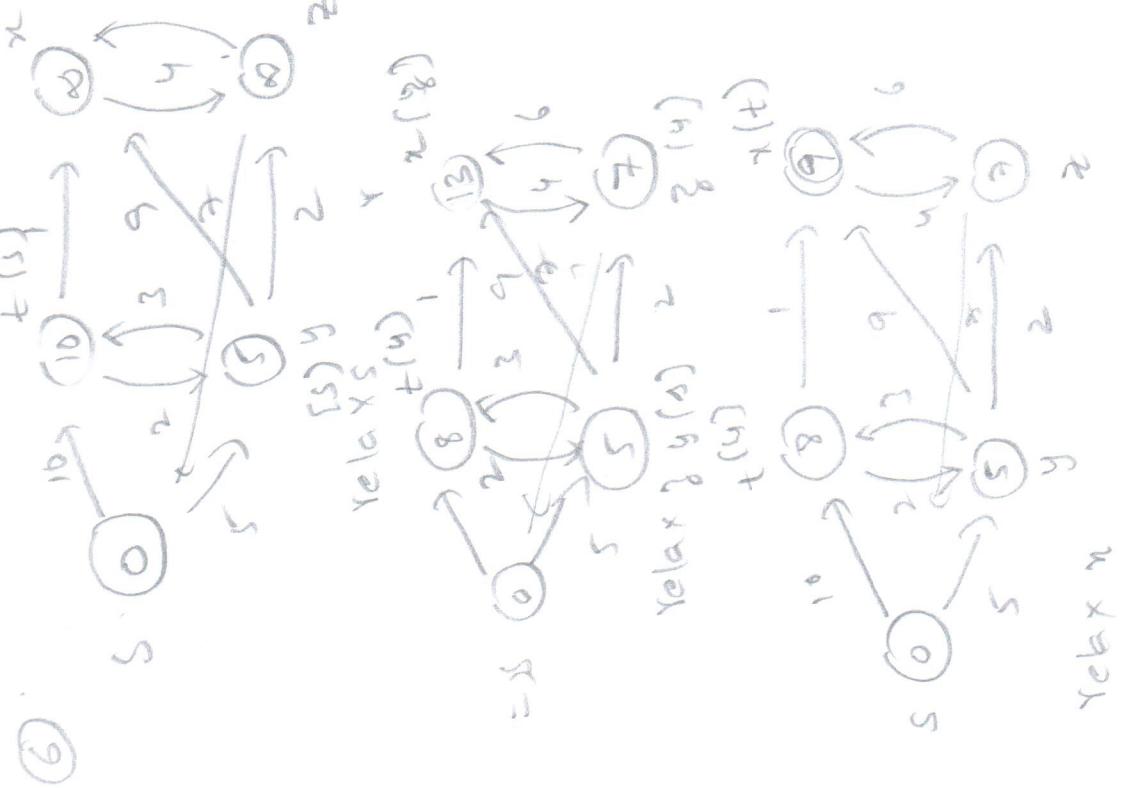
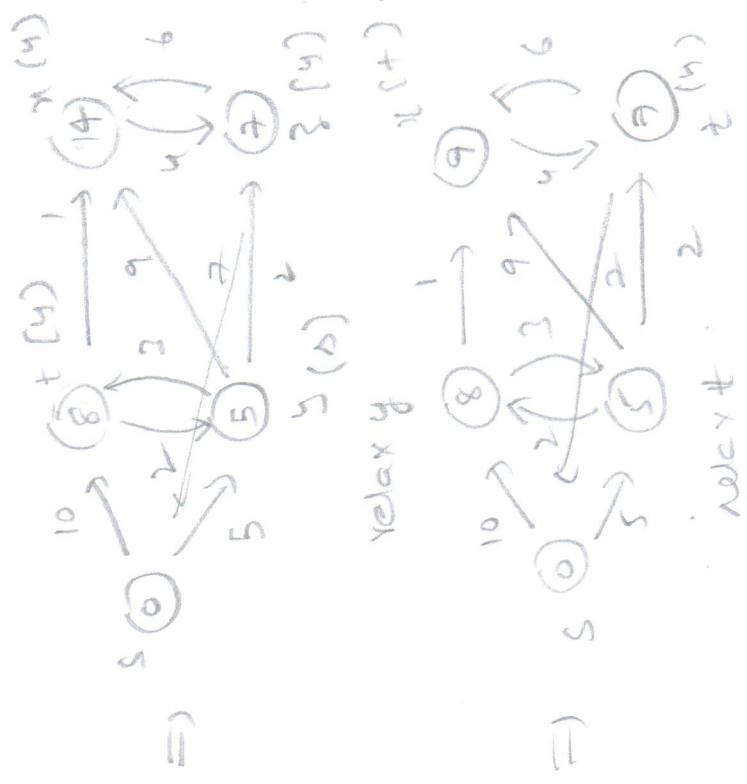
h:15 0 1 0

i:27 0 0

j:9 0 1 1 1







3	5	1	2	3	1	4
8	10	5	2	2	1	6
9	8	5	14	7	1	5

	t	y	x	z
s	0	2	2	2
y	0	10	5	2
t	0	8	5	14
x	0	8	5	9
z	0	8	5	9
	0	8	5	7

⑥ Time complexity for min heap $\frac{O((V+E) \log V)}{\text{OK}}$ ✓

③

$$1. t = e(i,r) + e(r+1,j) + w(i,j)$$

w

$$ij + e(i,j)$$

$$2. e(i,j) = t \quad \checkmark$$

$$3. \text{root}(i,j) = r \quad \checkmark$$

i	0	1	2	3	4
P _i	0.12	0.08	0.15	0.18	
g _i	0.08	0.06	0.09	0.10	0.14

	0	1	2	3	4
1	0.08	0.26	0.43	0.68	1
2		0.06	0.22	0.48	0.8
3			0.09	0.24	0.66
4				0.10	0.42
5					0.14

$$w(1,1) = w(1,0) + P_1 + g_1$$

$$= 0.08 + 0.12 + 0.06 = 0.26$$

$$w(2,2) = w(2,1) + P_2 + g_2$$

$$= 0.06 + 0.08 + 0.09 = 0.23$$

$$w(3,3) = w(3,2) + P_3 + g_3$$

$$= 0.09 + 0.15 + 0.10 = 0.34$$

$$w(4,4) = w(4,3) + P_4 + g_4$$

$$= 0.10 + 0.18 + 0.14 = 0.42$$

$$w(1,2) = w(1,1) + P_2 + g_2$$

$$= 0.26 + 0.08 + 0.09 = 0.43$$

$$w(2,3) = 0.23 + 0.15 + 0.10 = 0.48$$

$$w(3,4) = 0.34 + 0.18 + 0.14 = 0.66$$

$$w(1,3) = 0.43 + 0.15 + 0.10 = 0.68$$

$$w(2,4) = 0.48 + 0.18 + 0.14 = 0.79$$

$$w(3,4) = 0.66 + 0.18 + 0.14 = 1$$

root = 1

$$e(1,1) = e(1,0) + e(2,1) + w(1,1)$$

$$= 0.08 + 0.06 + 0.26 = 0.4$$

r = 2

$$e(2,2) = e(2,1) + e(3,2) + w(2,2)$$

$$= 0.06 + 0.09 + 0.22 = 0.37$$

$$e(3,3) = e(3,2) + e(4,3) + w(3,3)$$

$$= 0.09 + 0.10 + 0.24 = 0.43$$

	0	1	2	3	4
1	0.08	0.4	0.89	1.61	2.55
2		0.06	0.37	0.96	1.87
3			0.09	0.53	1.33
4				0.10	0.66
5					0.14

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

OK

$$e(4,4) = e(2,3) + e(5,4) + w(4,4)$$

$$= 0.10 + 0.14 + 0.42$$

$$= 0.66$$

$e(1,2)$

$$e(1,2) = \begin{aligned} & e(1,1) + e(2,2) + w(1,2) \\ & = 0.08 + 0.19 + 0.43 = \underline{0.89} \end{aligned}$$

$y=2$

$$e(1,2) + e(3,2) + w(1,2) \\ = 0.4 + 0.09 + 0.43 = 0.92$$

$e(2,1) : y=2$

$$e(2,1) + e(3,2) + w(2,1) \\ = 0.06 + 0.52 + 0.48 = 1.07$$

$y=2$

$$e(2,2) + e(4,2) + w(2,2) \\ = 0.37 + 0.19 + 0.48 = \underline{0.96}$$

$e(3,4) : y=3$

$$e(3,2) + e(4,4) + w(3,4) \\ = 0.09 + 0.66 + 0.66 = 1.4$$

$y=4$

$$e(3,3) + e(5,4) + w(3,4) \\ = 0.153 + 0.19 + 0.66 = \underline{1.33}$$

$y=1$

$$e(1,3) = e(1,0) + e(2,3) + w(1,3) \\ = 0.08 + 0.96 + 0.68 = 1.72$$

$$e(1,2) + e(3,3) + w(1,2) \\ = 0.4 + 0.52 + 0.68 = \underline{1.61}$$

$$e(1,2) + e(4,3) + w(1,2) \\ = 0.89 + 0.19 + 0.68 = \underline{1.77}$$

$e(2,4) : y=2$

$$e(2,1) + e(3,4) + w(2,4) \\ = 0.06 + 0.19 + 0.8 = 2.19$$

$$e(2,2) + e(4,4) + w(2,4) \\ = 0.37 + 0.66 + 0.8 = \underline{1.84}$$

$y=4$

$$e(2,3) + e(5,4) + w(2,4) \\ = 0.96 + 0.19 + 0.8 = \underline{1.97}$$

$y=1$

$e(1,4) : y=1$

$$e(1,0) + e(2,4) + w(1,4) \\ = 0.08 + 1.86 + 1 = 2.92$$

$y=2$

$$e(1,1) + e(3,4) + w(1,4) \\ = 0.4 + 1.32 + 1 = 2.73$$

$y=2$

$$e(1,2) + e(4,4) + w(1,4) \\ = 0.89 + 0.66 + 1 = \underline{2.55}$$

$y=4$

$$e(1,3) + e(5,4) + w(1,4) \\ = 1.61 + 0.19 + 1 = \underline{2.88}$$

