INT102-Assessment 1

sID:2257475 Name: Ruiyang Wu Date:2024-04-11

Question 1 (15 marks)

Consider the following function: $f(n) = 9n + 3n^2 + 2n \log n + 3\sqrt{n}$

(a) State the order of magnitude (in Big-O notation) of the function. **(5 marks)**

(b) Prove that the function *f(n)* is of the order of magnitude as you stated above. **(10 marks)**

(a) $\because 3n^2 > 2n\log n > 9n > 3\sqrt{n}$

$\therefore$ the order of $3n^2$ is n^2.

$\therefore$ The order of magnitude is $O(n^2)$

(b) To prove that $f(n)$ is $O(n^2)$, we show that there exist a constant c and n_0 that for any integer $n \geq n_0$

$f(n) = 3n^2 + 2n\log n + 9n + 3\sqrt{n} \leq cn^3$

$3n^2 \leq 3n^2$ for $\forall n$

$2n\log n \leq 2n^2$ for $\forall n \geq 1$

$9n \leq 9n^2$ for $\forall n$

$3\sqrt{n} \leq 3n^2$ for $\forall n \geq 1$

As a result, $f(n) \leq 17n^2$ for $\forall n \geq 1$

$\because$ 17 is a constant,

$\therefore$ the function $f(n)$ is $O(n^2)$.

Q.E.D.

Question 2 (30 marks)

The time complexity of the merge sort algorithm can be described by the following recurrence for T(n).

$$T(n)=\begin{cases}1 & \text{if} \quad n=1 \\ 2\times T(n/2)+n & \text{if} \quad n>1\end{cases}$$

(a) Explain the recurrence in terms of Divide and Conquer design technique. **[15 marks]**
(b) Prove that *T(n) = O(n log n)* (Guess: T(n) ≤ 2 n log n). **[15 marks]**

(a) In merge sort, Firstly, we divide a sequence of total length n into 2 smaller sequences recursively. We need to divide Log(n) times until there is only 1 number in each sequence so that we can conquer the problem easily

Then, we only need to compare each 2 sequences and merge them into bigger Ordered sequences.
It is easy to conquer since the time complexity of the merge of 2 Ordered sequences is only $O(n)$.

Finally, after $\log(n)$ times of merging (which need n times of comparing each time), the total time complexity of merge sort is $O(n \log n)$ instead of $O(n^2)$ in selection sort.

This is the charm of Divide & Conquer.

[3 1 4 6 5 8 7 2]
[3 1 4 6] [5 8 7 2]
[3 1] [4 6] [5 8] [7 2]
[3] [1] [4] [6] [5] [8] [7] [2]

Divide

[1,3] [4,6] [5,8] [2,7]
[1,3,4,6] [2,5,7,8]
[1,2,3,4,5,6,7,8]

$\boxed{\log n \times n}$ Conquer

(At the cost of memory $O(n)$)

(b) For an array of size n we have the recurrence relation:

$$T(n) = \begin{cases} 1 & , \text{if } n = 1 \\ 2T(\frac{n}{2}) + n & , \text{if } n > 1 \end{cases}$$ Guess: $T(n) \leq 2n\log n$.

① Base case: when $n=2$, $L.H.S = T(2) = 2T(1) + 2 = 4$

$R.H.S. = 2 \times 2\log 2 = 4$, $L.H.S. \leq R.H.S$, which is true

② Assume it is true for all $n' < n$,

$\because T(n) \leq 2n\log n$.

$\therefore T(\frac{n}{2}) \leq n\log\frac{n}{2} = n(\log n - 1) = n\log n - n$.

$\therefore T(n) = 2(n\log n - n) + n$.

$= 2n\log n - n \leq 2n\log n$.

$\therefore$ The time complexity of $T(n)$ is $O(n\log n)$.

Q.E.D.

Question 3 (15 marks)

Given the Bubble sort algorithm as below:

```
ALGORITHM BubbleSort(A[0..n − 1])
//Sorts a given array by bubble sort
//Input: An array A[0..n − 1] of orderable elements
//Output: Array A[0..n − 1] sorted in ascending order
for i=0 to n − 2 do
    for j = n-1 downto i+1 do
        if A[j ]<A[j-1 ] swap A[j ] and A[j - 1]
```

(a) What is the number of swapping operations needed to sort the numbers A[0..5]=[2，4，6，2，4，6] in ascending order using the Bubble sort algorithm? **(6 marks)**
(b) What is the number of key comparisons needed to sort the numbers A[0..5]= [3，4，5，3，4，5] in ascending order using the Bubble sort algorithm? **(9 marks)**

(a)

loop No.	Array.	
0	[2 4 6 2 4 6].	→ 2 swap.
1	[2 2 4 6 4 6]	→ 1 swap.
2	[2 2 4 4 6 6]	
3	Ordered.	
4		

∴ Number of swapping operations = 3.

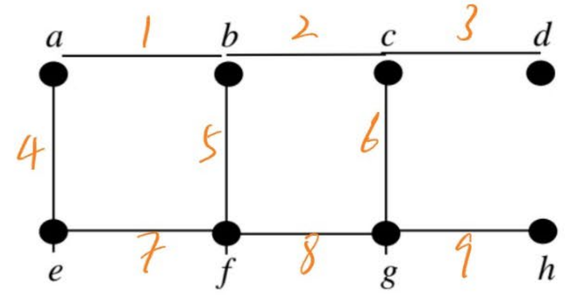
(b) n of A[0...5] = 6.
key comparision of each loop = from (6-1) to i

∴ key comparision $= \sum_{i=0}^{n-2} (n-1-i)$.

$$= \sum_{i=0}^{4} (5-i) = 15$$

∴ The number of ~~key~~ key comparision = 15

Question 4 (20 marks)
Consider the following graph G.



(a) Give the adjacency matrix and adjacency list of the graph *G*. **(10 marks)**
(b) Give the incidence matrix and incidence list of the graph *G*. **(10 marks)**

(a) adjacency matrix:

$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array}
\begin{array}{c}
\begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

adjacency list:

a → b → e
b → a → c → f
c → b → d → g
d → c
e → a → f
f → b → e → g
g → c → f
h → g

(b) incidence matrix:

The name of edges are marked in Graph G.

$$
\begin{array}{c}
\\
1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array}\\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
$$

incidence list:

1 → a → b

2 → b → c

3 → c → d

4 → a → e

5 → b → f

6 → c → g

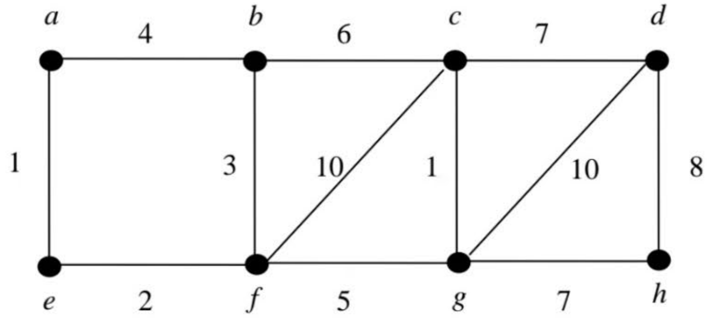
7 → e → f

8 → f → g

9 → g → h

Question 5 (20 marks)

Consider the following graph G. The label of an edge is the cost of the edge.



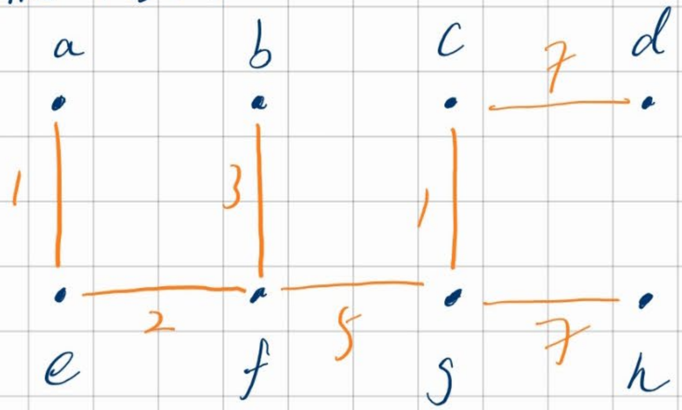
(a) Using *Prim's* algorithm, draw a *minimum spanning tree* (MST) of the graph Also write down the change of the priority queue step by step and the order in which the vertices are selected. Is the MST drawn unique? (i.e., is it the one and only MST for the graph?) **[7 marks]**

(b) Using *Kruskal's* algorithm, draw a *minimum spanning tree* (MST) of the graph G. Write down the order in which the edges are selected. Is the MST drawn unique? (i.e., is it the one and only MST for the graph?) **(7 marks)**

(c) Referring to the same graph above, find the shortest paths from the vertex ***a*** to *all* other vertices in the graph G using *Dijkstra's* algorithm. Show the changes of the priority queue step by step and give the order in which edges are selected. **(6 marks)**

N.B. There may be more than one solution. You only need to give one of the solutions.

(a) Prim-MST:

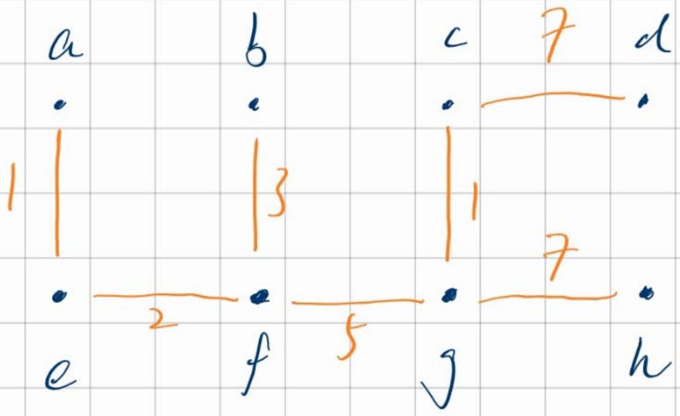


Priority queue - Prim								
Order selected	a(0,-)	b(-,∞)	c(-,∞)	d(-,∞)	e(-,∞)	f(-,∞)	g(-,∞)	h(-,∞)
a(0,-)		b(a,4)	c(-,∞)	d(-,∞)	e(a,1)	f(-,∞)	g(-,∞)	h(-,∞)
e(a,1)		b(a,4)	c(-,∞)	d(-,∞)		f(e,2)	g(-,∞)	h(-,∞)
f(e,2)		b(f,3)	c(f,10)	d(-,∞)			g(f,5)	h(-,∞)
b(f,3)			c(b,6)	d(-,∞)			g(f,5)	h(-,∞)
g(f,5)			c(g,1)	d(g,10)				h(g,7)
c(g,1)				d(c,7)				h(g,7)
d(c,7)								h(g,7)
h(g,7)								

The MST of Graph G is unique.

(b) Kruskal - MST:

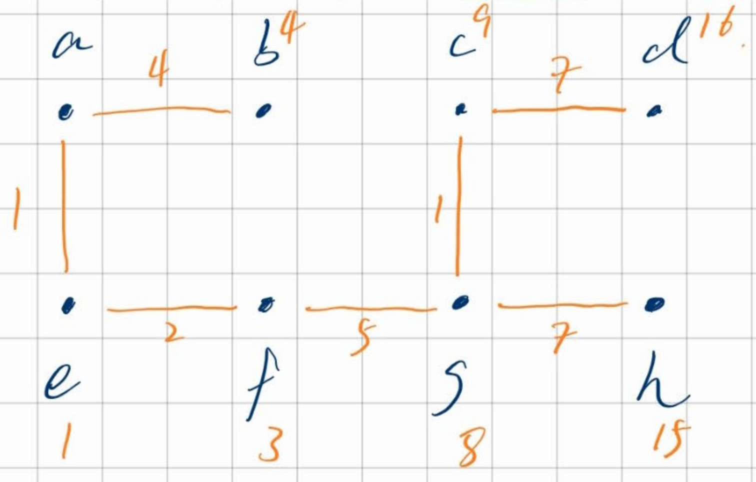
Kruskal-Order		
edge	**weight**	**select order**
a-e	1	1
c-g	1	2
e-f	2	3
b-f	3	4
a-b	4	cyclization
f-g	5	5
b-c	6	cyclization
c-d	7	6
g-h	7	7
d-h	8	cyclization
c-f	10	cyclization
d-g	10	cyclization



The MST of Graph G is unique.

(c)

Priority queue - Dijkstra								
Order selected	a(0,-)	b(-,∞)	c(-,∞)	d(-,∞)	e(-,∞)	f(-,∞)	g(-,∞)	h(-,∞)
1		b(a,4)-4	c(-,∞)	d(-,∞)	e(a,1)-1	f(-,∞)	g(-,∞)	h(-,∞)
2		b(a,4)-4	c(-,∞)	d(-,∞)		f(e,2)-3	g(-,∞)	h(-,∞)
3		b(a,4)-4	c(f,10)-13	d(-,∞)			g(f,5)-8	h(-,∞)
4			c(b-6)-10	d(-,∞)			g(f,5)-8	h(-,∞)
5			c(g,1)-9	d(g,10)-18				h(g,7)-15
6				d(c,7)-16				h(g,7)-15
7				d(c,7)-16				



Shortest Path:

a→b : a→b = 4

a→c : a→e→f→g→c = 9

a→d : a→e→f→g→c→d = 16

a→e : a→e = 1

a→f : a→e→f = 3

a→g : a→e→f→g = 8

a→h : a→e→f→g→h = 15.

wish you have a good day ! :)