Module: INT102 **Assignment 1**

1. Assessment

The tasks contribute 10% to the overall assessment of INT102

2. Submission

Please complete the assessment tasks using Microsoft Word and submit it in PDF via Learning.

3. Deadline

19-April- 2021, Monday 17:30.

Name: Lincheng Shi (施林呈）
Student_ID: 1927978

Question 1

1. $f(n) = 3n + n^2 + 2n \log n + 2\sqrt{n}$ is $O(n^2)$.

2. Since $n \le n^2$, $\sqrt{n} \le n^2$ and $\log n \le n$ for all $n \ge 1$,

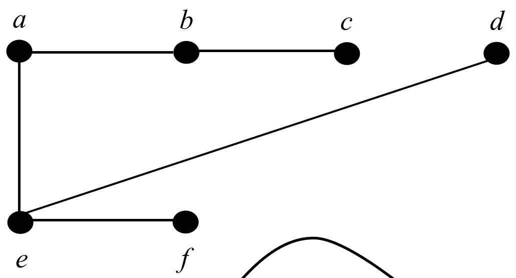
   we have $n \log n \le n^2$ and

   $3n + n^2 + 2n \log n + 2\sqrt{n} \le 3n^2+n^2+2n^2+2n^2=8n^2$ for all $n \ge 1$.

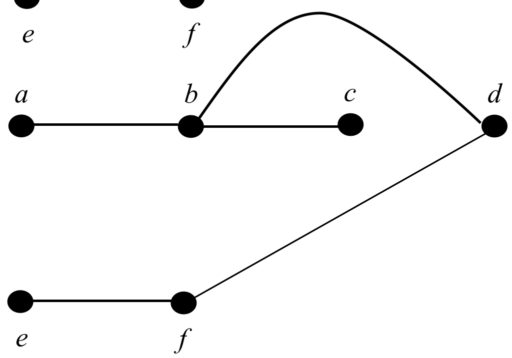
   Therefore, by definition, $f(n) = 3n + n^2 + 2n \log n + 2\sqrt{n}$ is $O(n^2)$.

Question 2

1.



2.



Question 3

3 4 5 3 4 5
3 4 5 3 4 5
3 4 3 5 4 5
3 3 4 5 4 5
3 3 4 5 4 5

3 3 4 5 4 5
3 3 4 4 5 5
3 3 4 4 5 5
3 3 4 4 5 5

3 3 4 4 5 5
3 3 4 4 5 5
3 3 4 4 5 5

3 3 4 4 5 5
3 3 4 4 5 5

3 3 4 4 5 5

1. The number of swapping operations is 3.
2. 5+4+3+2+1=15
   The number of key comparisons is 15.

Question 4

1. $f_1[j]=\begin{cases} a_{1,1}+t_{0,1} & \text{, if } j=1 \\ \min(f_1[j-1]+a_{1,j},\ f_2[j-1]+t_{2,j-1}+a_{1,j}) & \text{, if } j>1 \end{cases}$

   $f_2[j]=\begin{cases} a_{2,1}+t_{0,2} & \text{, if } j=1 \\ \min(f_2[j-1]+a_{2,j},\ f_1[j-1]+t_{1,j-1}+a_{2,j}) & \text{, if } j>1 \end{cases}$

   $f^*=\min(f_1[n]+t_{1,n},\ f_2[n]+t_{2,n})$
2.

j	$f_1[j]$	$f_2[j]$
1	4	6
2	12	8
3	11	14
4	16	13

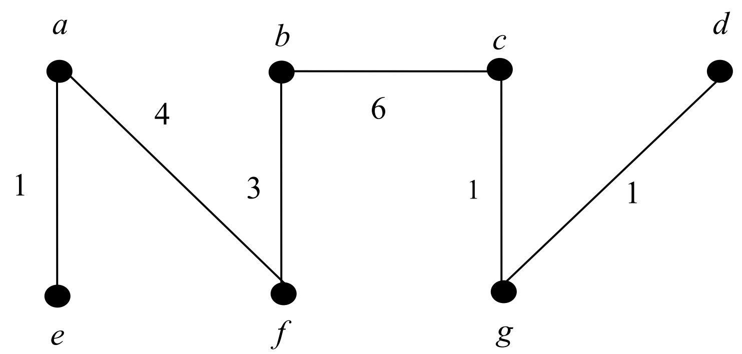
3.set $f_1[1]=a_{1,1}+t_{0,1}$
  set $f_2[1]=a_{2,1}+t_{0,2}$
  for j=2 to n do
  begin
    set $f_1[j]=\min(f_1[j-1]+a_{1,j},\ f_2[j-1]+t_{2,j-1}+a_{1,j})$
    set $f_2[j]=\min(f_2[j-1]+a_{2,j},\ f_1[j-1]+t_{1,j-1}+a_{2,j})$
end
set $f^*=\min(f_1[n]+t_{1,n},\ f_2[n]+t_{2,n})$
Time complexity is O(n).

3. start: ①2 ②3
   $S_{1,1}$: 2+2
   $S_{2,1}$: 3+3
   $S_{1,2}$: ①4+8 ②6+1+8 ①<②
   $S_{2,2}$: ①3+3+4 ②4+4 ①>②
   $S_{1,3}$: ①12+2 ②8+1+2 ①>②
   $S_{2,3}$: ①12+6 ②8+6 ①>②
   $S_{1,4}$: ①11+5 ②14+1+5 ①<②
   $S_{2,4}$: ①11+1+1 ②14+1 ①<②
   end: ①16+3 ②13+1 ①>②
   start→$S_{1,1}$→$S_{1,1}$→$S_{2,2}$→$S_{1,3}$→$S_{2,4}$→end

Question 5

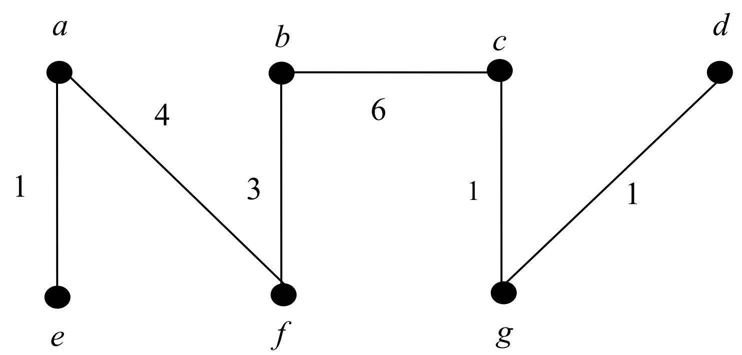
1.

Order selected	a(0,-)	b(-,∞)	c(-,∞)	d(-,∞)	e(-,∞)	f(-,∞)	g(-,∞)
a(0,-)		b(-,∞)	c(-,∞)	d(a,9)	e(a,1)	f(a,4)	g(-,∞)
e(a,1)		b(-,∞)	c(-,∞)	d(a,9)		f(a,4)	g(-,∞)
f(a,4)		b(f,3)	c(f,10)	d(a,9)			g(-,∞)
b(f,3)			c(b,6)	d(a,9)			g(-,∞)
c(b,6)				d(a,9)			g(c,1)
g(c,1)				d(g,1)			
d(g,1)							

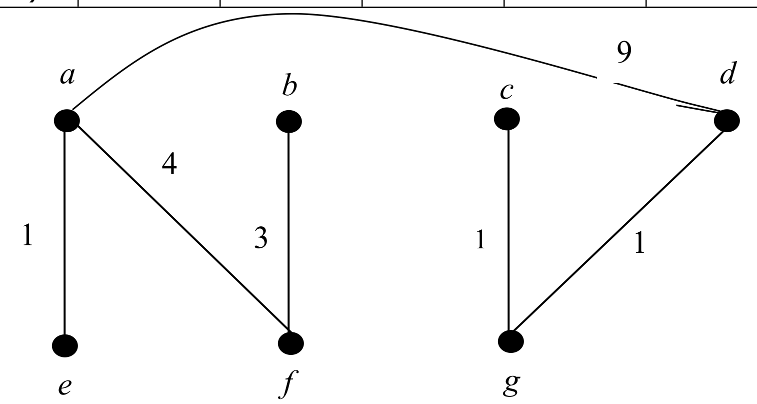


2.

(a,e)	1/
(c,g)	1/
(d,g)	1/
(b,f)	3/
(a,f)	4/
(b,c)	6/
~~(a,d)~~	~~9~~
~~(c,f)~~	~~10~~



Order selected	a(0,-)	b(-,∞)	c(-,∞)	d(-,∞)	e(-,∞)	f(-,∞)	g(-,∞)
a(0,-)		b(-,∞)	c(-,∞)	d(a,9)	e(a,1)	f(a,4)	g(-,∞)
e(a,1)		b(-,∞)	c(-,∞)	d(a,9)		f(a,4)	g(-,∞)
f(a,4)		b(f,7)	c(f,14)	d(a,9)			g(-,∞)
b(f,7)			c(b,13)	d(a,9)			g(-,∞)
d(a,9)			c(b,13)				g(d,10)
g(d,10)			c(g,11)				
c(g,11)							



Question 6

1. Algorithm: Exponentiation

//compute a^n where n is a positive integer

//Input: a number a and a positive integer n

//Output: the reult of a^n

IF n is even

$a^n=a^{n/2}*a^{n/2}$

IF n is odd

$a^n=a^{(n-1)/2}*a^{(n-1)/2}*a$

RETURN a^n

2. T(n)=1 , n=1

T(n/2)+1, n>1

Assume that $n=2^k$

T(n)= T(n/2) + 1

= $T(n/2^2)+1+1$

= $T(n/2^3)+1+1+1$

……

= $T(n/2^k)+1+...+1+1+1$

= T(1)+k

= 1+log n

3. The time complexity of this algorithm is O(log n), but the time complexity of brute-force algorithm is O(n). Because O(log n)<O(n), divide-and conquer algorithm is more efficient.

Question 7

1. The definition of line i and the j-th point is arr[i][j]. There are n lines in total. To find out the minimum value of arr[1][1], we need to compare arr[2][1] and arr[2][2] which is smaller. To get the minimum value of arr[i][j], we need to compare arr[i+1][j] and arr[i+1][j+1] which is smaller.
2. arr[i][j]=min(arr[i+1][j],arr[i+1][j+1])+arr[i][j]
3. for i=n-1 downto 1 do

```
        for j=1 to i do
          begin arr[i][j]=min(arr[i+1][j],arr[i+1][j+1])+arr[i][j]
    end
    return arr[1][1]
```

   Time complexity: (n-1)+(n-2)+(n-3)+...+1=n(n-1)/2

   So the time complexity is $O(n^2)$.

4. arr[4][1]<arr[4][2]   arr[3][1]=3+2=5
   arr[4][2]<arr[4][3]   arr[3][2]=6+3=9
   arr[4][3]<arr[4][4]   arr[3][3]=8+4=12
   arr[3][1]<arr[3][2]   arr[2][1]=5+5=10
   arr[3][2]<arr[3][3]   arr[2][2]=9+4=13
   arr[2][1]<arr[2][2]   arr[1][1]=10+3=13

Thus, the smallest sum of the above minimum-sum descent problem is 13.