Q1.

a. $O(n^2)$

b. To prove that $f(n) = 7n + 3n^2 + 2n\log n + 2$ is $O(n^2)$, we show that there exists a constant C and n_0 such that for any integer $n \ge n_0$, $f(n) \le Cn^2$

Since $7n \le 7n^2 \quad \forall n \ge 1$

$2n\log n \le 2n^2 \quad \forall n \ge 2$

$2 \le 2n^2 \quad \forall n \ge 1$

So $f(n) = 7n + 3n^2 + 2n\log n + 2 \le 7n^2 + 2n^2 + 2n^2 + 3n^2 = 14n^2$

As a result, $\forall n \ge 2$, $f(n) \le 14n^2$, so by definition $f(n) = O(n^2)$

Q2.

a. Divide and Conquer

b. Initialize $l = 0 \quad m = 7$.

Iteration 1: $m = \lfloor \frac{0+7}{2} \rfloor = 3$, $A[3] = 49 < 99 = k$, so update $l = 4$

Iteration 2: $m = \lfloor \frac{4+7}{2} \rfloor = 5$, $A[5] = 55 < 99 = k$, so update $l = 6$

Iteration 3: $m = \lfloor \frac{6+7}{2} \rfloor = 6$, $A[6] = 99 = k$, return $m = 6$

So the output is 6

c. After first comparison, the search interval is reduced to $\frac{n}{2}$;

After second comparison, the search interval is reduced to $\frac{n}{4}$;

After kth comparison, the search interval is $\frac{n}{2^k}$;

The process stops when the search interval has been reduced to only one element, which means $\frac{n}{2^k} = 1$, so $2^k = n \Rightarrow k = \log_2 n$

Since each iteration uses a constant amount of work, so the ovellall run time is proportional to $\log_2 n$.

Hence the time complexity is $O(\log n)$.

Q3.

a. when $i=0$, compare with indices 1, 2, 3, 4, 5 $\longrightarrow$ 5 comparisons

when $i=1$, compare with indices 2, 3, 4, 5 $\longrightarrow$ 4 comparisons

when $i=2$, compare with indices 3, 4, 5 $\longrightarrow$ 3 comparisons

when $i=3$, compare with indices 4, 5 $\longrightarrow$ 2 comparisons

when $i=4$, compare with indices 5 $\longrightarrow$ 1 comparison

So Total comparisons $= 5+4+3+2+1 = 15$

b. ① compare 6 and 1, swap $\longrightarrow$ [1, 6, 2, 3, 4, 5]

②. compare 6 and 2, swap $\longrightarrow$ [1, 2, 6, 3, 4, 5]

③ compare 6 and 3, swap $\longrightarrow$ [1, 2, 3, 6, 4, 5]

④. compare 6 and 4, swap $\longrightarrow$ [1, 2, 3, 4, 6, 5]

⑤. compare 6 and 5, swap $\longrightarrow$ [1, 2, 3, 4, 5, 6]

The array has been sorted into the ascending order.

So the number of swapping operations is 5.

Q4.

a. The adjacency matrix:

	a	b	c	d	e	f
a	0	1	0	0	1	0
b	1	0	1	1	0	1
c	0	1	0	1	0	1
d	0	1	1	0	0	1
e	1	0	0	0	0	1
f	0	1	1	1	1	0

The adjacency list:

a → b → e

b → a → c → d → f

c → b → d → f

d → b → c → f

e → a → f

f → b → c → d → e

b. Let $e_1=(a,b)$, $e_2=(a,e)$, $e_3=(e,f)$, $e_4=(b,f)$, $e_5=(b,c)$, $e_6=(c,f)$, $e_7=(d,f)$,

$e_8=(c,d)$, $e_9=(b,d)$

The incidence matrix:

	a	b	c	d	e	f
e_1	1	1	0	0	0	0
e_2	1	0	0	0	1	0
e_3	0	0	0	0	1	1
e_4	0	1	0	0	0	1
e_5	0	1	1	0	0	0
e_6	0	0	1	0	0	1
e_7	0	0	0	1	0	1
e_8	0	0	1	1	0	0
e_9	0	1	0	1	0	0

The incidence list:

e_1 → a → b

e_2 → a → e

e_3 → e → f

e_4 → b → f

e_5 → b → c

e_6 → c → f

e_7 → d → f

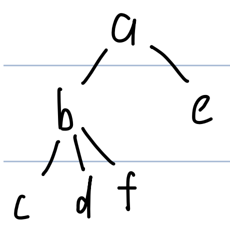
e_8 → c → d

e_9 → b → d

c.

Order of BFS traversal: a → b → e → c → d → f

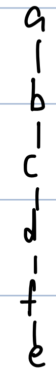
BFS Tree:



d.

Order of DFS traversal: a → b → c → d → f → e

DFS Tree:

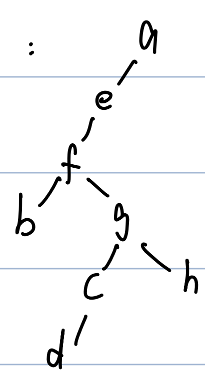


Q5

a. The order in which the vertices are selected is as follows: a, e, f, b, g, c, d, h

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:



The MST drawn is not unique

b. Sorting edges according to their weights in increasing order:

(a,e) 1

(c,g) 1

(e,f) 2

(b,f) 3

~~(a,b) 4~~

(f,g) 5

~~(b,c) 6~~

(c,d) 7

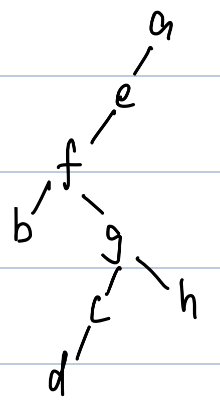
(g,h) 7

~~(d,h) 8~~

~~(c,f) 10~~

~~(d,g) 10~~

The MST:



The MST is not unique

c. Selected order of edges: (a,e), (e,f), (a,b), (f,g), (g,c), (g,h), (c,d)

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,3)	g(-,∞)	h(-,∞)
f(e,3)		b(a,4)	c(f,13)	d(-,∞)			g(f,8)	h(-,∞)
b(a,4)			c(b,10)	d(-,∞)			g(f,8)	h(-,∞)
g(f,8)			c(g,9)	d(g,18)				h(g,15)
c(g,9)				d(c,16)				h(g,15)
h(g,15)				d(c,16)				
d(c,16)								

The shortest paths :