Question 1

1. Pattern: AGTAA Shift table:

A	G	C	T
1	3	5	2

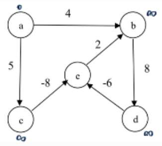
2. Text:

```
 C    C
AGCCGTGC
A GTAA
   AGTAA
```

comparison time: 2

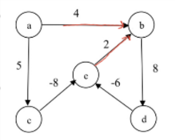
Question 2

The 1st iteration



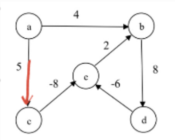
The 1st iteration

vertice	a	b	c	d	e
cost	0	∞	∞	∞	∞



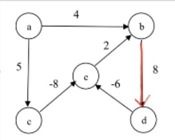
vertice	a	b	c	d	e
cost	0	∞	∞	∞	∞
ab	0	4	∞	∞	∞
eb	0	∞	∞	∞	∞

d[A]+W(A,B)=4



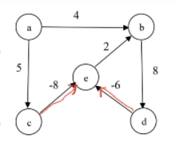
vertice	a	b	c	d	e
cost	0	∞	∞	∞	∞
ab	0	4	∞	∞	∞
eb	0	∞	∞	∞	∞
ac	0	4	5	∞	∞

d[A]+w(A,C)=5



vertice	a	b	c	d	e
cost	0	∞	∞	∞	∞
ab	0	4	∞	∞	∞
eb	0	∞	∞	∞	∞
ac	0	4	5	∞	∞
bd	0	4	5	12	∞

d[b]+w(b,d)=4+8=12



vertice	a	b	c	d	e
cost	0	∞	∞	∞	∞
ab	0	4	∞	∞	∞
eb	0	∞	∞	∞	∞
ac	0	4	5	∞	∞
bd	0	4	5	12	∞
ce	0	4	5	12	-3
de	0	4	5	12	6
i=1	0	4	5	12	-3

d[c]+w(c,e)=5-8=-3

d[d]+w(d,e)=12-6=6

The 2nd iteration:

vertice	a	b	c	d	e
cost	0	4	5	12	-3
eb	0	-1	5	12	-3
bd	0	-1	5	7	-3
∴ i=2	0	-1	5	7	-3

d[e]+w(e,b)=-3+2=-1

d[b]+w(b,d)=-1+8=7

The 3rd iteration: No more updating.

Question 3

1.

		0	1 A	2 G	3 A	4 C	5 C	6 T
0		0	0	0	0	0	0	0
1	G	0	0	1	1	1	1	1
2	A	0	1	1	2	2	2	2
3	G	0	1	2	2	2	2	2
4	T	0	1	2	2	2	2	3

∴ the length is 3

2. ∴ the longest common subsequence is: GAT

		0	1 A	2 G	3 A	4 C	5 C	6 T
0		0	0	0	0	0	0	0
1	G	0	↑0	↖1	←1	←1	←1	←1
2	A	0	↖1	↑1	↖2	←2	←2	←2
3	G	0	↑1	↖2	↑2	←2	←2	←2
4	T	0	↑1	↑2	↑2	↑2	↑2	↑3

```
①
- - GA_GT
A C-ATGT

②
GA - - _GT
-ACATGT ✓
```

Question 4

1. a) complete the table using the formula:

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + S(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \end{cases}$$

	A	C	G	T
A	1	-3	-2	-3
C	-3	1	-3	-2
G	-2	-3	1	-3
T	-3	-2	-3	1

and set $F(0,0)=0$,

$F(i,0) = i \times d$

$F(0,j) = j \times d$

d is gap penalty = -1

		G	A	G	T
	0	←-1	←-2	←-3	←-4
A	↑-1	←-2	↖0	←-1	←-2
C	↑-2	←-3	↑-1	←-2	←-3
A	↑-3	←-4	↖↑-2	←-3	←-4
T	↑-4	←-5	↑-3	←-4	↖-2
G	↑-5	↖-3	←-4	↖-2	←-3
T	↑-6	↑-4	←-5	↑-3	↖-1

b) there are 4:

```
① G_A_GT
  ACATGT

② GA___GT
  _ACATGT ✓

③ -GA_GT
  ACATGT

④ -G-A-GT
  A-CATGT
```

垂直当前序列的箭头表示那个轴的空格

2. a) Complete the following table using the formula:

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + S(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

and set $F(0,0) = 0$

		G	A	G	T
	0	0	0	0	0
A	0	0	↖ 1	0	0
C	0	0	0	0	0
A	0	0	↖ 1	0	0
T	0	0	0	0	↖ 1
G	0	↖ 1	0	↖ 1	0
T	0	0	0	0	↖ 2

b). there are only 1 condition:

G T

G T

Question 5:

To reduce complexity, assume that e is before d

	a	b	c	d	e
a	0	4	5	2	1
b	4	0	4	3	1
c	5	4	0	1	8
d	2	3	1	0	6
e	1	1	8	6	0

Node 0 : $lb = \lceil (1+2+1+3+1+4+1+2+1+1)/2 \rceil = 9$

Node 1 : $lb = \lceil (4+1+4+1+4+1+2+1+2)/2 \rceil = 10$

Node 2 : $lb = \lceil (5+1+5+1+2+4+3)/2 \rceil = 11$

Node 3: ignored as b is not before

Node 4: $lb = \lceil (3+2+4+3+5)/2 \rceil = 9$

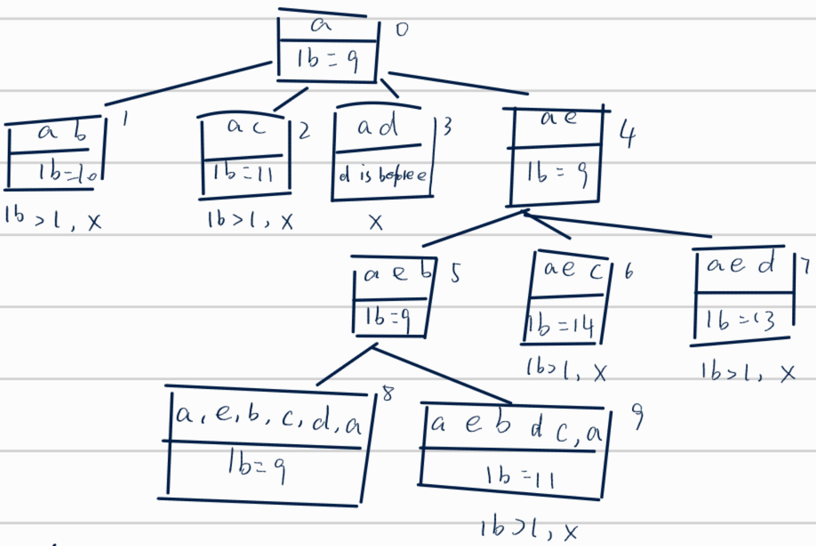
Node 5: $lb = \lceil (1+2+2+4+3+5)/2 \rceil = 9$

Node 6: $lb = \lceil (3+9+9+3+4)/2 \rceil = 14$

Node 7: $lb = \lceil (3+7+3+5+7)/2 \rceil = 13$

Node 8: $lb = (1+1+4+1+2) = 9$

Node 9: $lb = 1+1+3+1+5 = 11$



Question 6:

1. if $P = NP$

∴ All the NP Problems can be solved in polynomia algorithm

Since all NP problems can be reduced to NPC algorithm,

∴ NPC problems can also be solved in polynomia time

∴ $P = NP = NPC$ is right

2. ∵ $P = NP$

∴ $NPC = NP$

3. If $P \neq NP$,

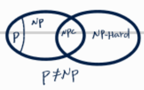
∵ $P \neq NP$

∴ there are some problems in NP that can not be solved by polynomia algorithm

∵ these problems may not be NPC either, but still in NP

that is, $NPC \subset NP$ but $NPC \neq NP$

∴ $NP \neq P \cup NPC$



∵ $NP \neq P$

∴ $P \cap NPC = \{\}$

∴ 3 is partly contradictory

4. ∵ $NP \neq P$

∴ $P \cap NPC = \{\}$

4 is wrong

5. ∵ $NP \neq P$

∴ $P \cap NPC = \{\}$

5 is right