Q1.

1. Procedure $F(n)$

```
    if n==0 or n==1 or n==2 or n==3 then
        return 1
    else
        return F(n-1) + F(n-2) + F(n-3)
```

2. $O(3^n)$

Suppose $T(n)$ is the required time of calculating $f(n)$.

Then $T(n) = T(n-1) + T(n-2) + T(n-3) + O(1)$ and $T(0) = T(1) = T(2) = T(3) = O(1)$

The recursion tree expands by a 3 factor at each level, from level 0 having 1 node to level k having 3^k nodes. The height of the tree is $O(n)$ since slowest recursive call reduces n by 1. Thus the total number of nodes is : $1 + 3^1 + 3^2 + \cdots + 3^{n-1} = O(3^n)$.

In conclusion, the time complexity is $O(3^n)$

3. function $F(n)$

```
    Initialize A[0...n]
    A[0] ← 1, A[1] ← 1, A[2] ← 1, A[3] ← 1
    for i=4 to n do
        A[i] = A[i-1] + A[i-2] + A[i-3]
    return A[n]
```

4. $O(n)$

In each iteration, there is only addition operation. So each step is $O(1)$; The for loop operates n-3 times, so the overall time complexity is $O(n)$

Q2.

1. Suppose a_{ij} denotes assembly time at S_{ij}, t_{ij} denotes transfer time after S_{ij}

j	$f_1[j]$	$f_2[j]$
1	2	4
2	8	9
3	12	17
4	20	17

$$f_1[j] = \begin{cases} a_{11} & j=1 \\ \min(f_1[j-1] + a_{1j},\ f_2[j-1] + t_{2(j-1)} + a_{1j}) & j>1 \end{cases}$$

$$f_2[j] = \begin{cases} a_{21} & j=1 \\ \min(f_2[j-1] + a_{2j},\ f_1[j-1] + t_{1(j-1)} + a_{2j}) & j>1 \end{cases}$$

$$f^* = \min(f_1[n], f_2[n])$$

So $f_1[1] = 2$, $f_2[1] = 4$

$f_1[2] = \min(2+6,\ 4+1+6) = 8$ $\quad$ $f_2[2] = \min(4+5,\ 2+2+5) = 9$

$f_1[3] = \min(8+4,\ 9+2+4) = 12$ $\quad$ $f_2[3] = \min(9+8,\ 8+4+8) = 17$

$f_1[4] = \min(12+8,\ 17+1+8) = 20$ $\quad$ $f_2[4] = \min(17+4,\ 12+1+4) = 17$

2. $f^* = \min(f_1[4], f_2[4]) = \min(20, 17) = 17$

3. $S_{1,1} \longrightarrow S_{1,2} \longrightarrow S_{1,3} \xrightarrow{\text{transfer}} S_{2,4}$

Q3.

1. Shift table:

A	G	C	T
5	1	4	2

A: 5

C: 5-1-0 = 4

G: 5-1-3 = 1

T: 5-1-2 = 2

2.

```
AGCCGTGC        →    AGCCGTGC        →    AGCCGTGC
CGTGC                 CGTGC                    CGTGC
```

comparison: 1 1 5

The total number of comparison is: 1+1+5 = 7

Q4.

Vertices	a	b	c	d	e
Cost	0	∞	∞	∞	∞
Pre	-	-	-	-	-

The 1st iteration :

Vertices	a	b	c	d	e
Cost	0	∞	5	∞	4
Pre	-	-	a	-	a

Vertices	a	b	c	d	e
Cost	0	∞	5	∞	-3
Pre	-	-	a	-	c

Vertices	a	b	c	d	e
Cost	0	-1	5	∞	-3
Pre	-	e	a	-	c

Vertices	a	b	c	d	e
Cost	0	-1	5	4	-3
Pre	-	e	a	e	c

The 2nd iteration: no more updating

Check for negative weight cycle :

$d[e] + w(e,b) = d[b]$ $\quad$ $d[a] + w(a,c) = d[c]$ $\quad$ $d[a] + w(a,e) > d[e]$

$d[e] + w(e,d) = d[d]$ $\quad$ $d[c] + w(c,e) = d[e]$

So there is no negative weight cycle.

So the shortest path from vertex a: $\begin{cases} b: a-c-e-b \\ c: a-c \\ d: a-c-e-d \\ e: a-c-e \end{cases}$

Q5.

1.

		A	A	T	G
	0	-5 ←	-10 ←	-15 ←	-20
A	-5 ↑	2 ↖ ↑ ←	-3 ↖ ←	-8 ←	-13
G	-10 ↑	-3 ↑	-3 ↖ ↑ ←	-8	-6 ↖
C	-15	-8 ↑	-8 ↑	-8 ↖	-11 ↑

A A T G -
- A - G C

or

A A T G -
A - - G C

Score = -11

2.

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

optimal local alignment: A-A , G-G   local score = 2