1. Timing Analysis of the CAN Protocol — Part I

(1) What is the worst-case response time of μ0?

Iteration	LHS(Q0)	B0	RHS	Stop?
1	30	30	30	Yes

B0 = Max(C0,C1,C2) = 30
Q0 = B0 = 30
LHS = RHS , Stop!
worst-case response time = 10 + 30 = 40

(2) What is the worst-case response time of μ1?

Iteration	LHS (Q1)	B1	j	Q1 + τ	Tj	$\lceil \frac{Q_1+\tau}{T_j} \rceil$	Cj	RHS	Stop?
1	30	30	0	30.1	50	1	10	40	No
2	40	30	0	40.1	50	1	10	40	Yes

Iteration1:
B1 = Max(C1,C2) = 30
Q1 = B1 =30
LHS = 30 < 30+1*10 = 40 = RHS
Iteration2:
LHS = 40 = 30+1*10 = 40 = RHS , Stop!
worst-case response time = 30 + 40 = 70

(2) What is the worst-case response time of μ2?

Iteration	LHS (Q2)	B2	j	Q2 + τ	Tj	$\lceil \frac{Q_2+\tau}{T_j} \rceil$	Cj	RHS	Stop?
1	20	20	0 1	20.1	50 200	1 1	10 30	60	No
2	60	20	0 1	60.1	50 200	2 1	10 30	70	No
3	70	20	0 1	70.1	50 200	2 1	10 30	70	Yes

Iteration1:
B2 = C2 = 20
Q2 = B2 = 20
LHS = 20 < 20+1*10+1*30 = 60 = RHS

Iteration2:

LHS = 60 < 20+2*10+1*30 = 70 = RHS

Iteration3:

LHS = 70 = 20+2*10+1*30 = 70 = RHS , Stop!

worst-case response time = 20 + 70 = 90

2. Timing Analysis of the CAN Protocol — Part 2(coding)

worst-case response time 結果如下:

```
1.44
2.04
2.56
3.16
3.68
4.28
5.2
8.4
9.0
9.68
10.2
19.36
19.8
20.32
29.40
29.76
30.28
```

Source code 的部分:

```python
import numpy as np
messages = []
with open('input.dat', 'r', encoding='UTF-8') as file:
    numbers = float(file.readline())
    tau = float(file.readline())
    for data in file.readlines():
        data = data.strip()
        data = data.split()
        messages.append(data)
# print(numbers, tau, messages)
```

```python
def worstwaitingtime(number):
    num = number

    blocking_list = []
    for i in range(num,17):
        blocking_list.append(messages[i][1])
    block = float(max(blocking_list))
    # print(block)

    LHS = block
    Q = block
    RHS = 0
    while (1<2):
        RHS = 0
        for i in range(0,num):
            RHS += np.ceil((Q+tau)/float(messages[i][2]))*float(messages[i][1])
            print(RHS)
        RHS += block
        print(RHS)
        if LHS == RHS:
            break
        LHS = RHS
        Q = RHS
    return RHS+float(messages[num][1])
```

```python
if __name__  == "__main__":
    with open('output.txt', 'w', newline='') as f:
        for i in range(17):
            f.writelines(str(worstwaitingtime(i))+'\n')
```

3. Timing Analysis of TDMA-Based Protocols

(1)Please duplicate the schedule pattern

(4,10,1,2,6,7)

(2) Please duplicate the arriving times of frames in the frame arrival pattern

(0,3,5,6,10,13,15,16)

(3) Please duplicate the starting times of time slots in the schedule pattern

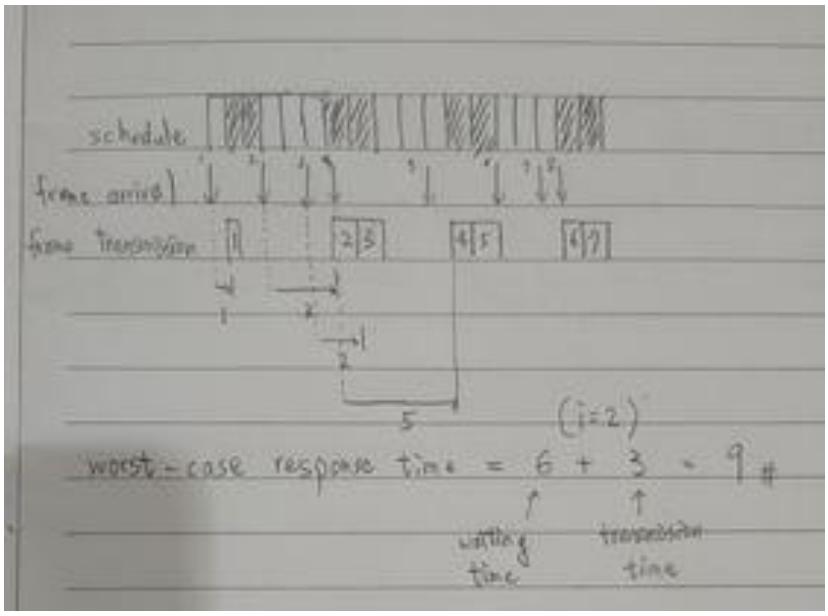
(1,2,6,7,11,12,16,17)

(4) Please complete the following table

k	$\max_{1\le j\le n}(s_{j+k}-s_j)$	=	$\min_{1\le i\le m}(a_{i+k-1}-a_i)$	=	(Column-3) − (Column-5)
1	$\max_{1\le j\le 4}(s_{j+1}-s_j)$		$\min_{1\le i\le 4}(a_i-a_i)$		
2	$\max_{1\le j\le 4}(s_{j+2}-s_j)$		$\min_{1\le i\le 4}(a_{i+1}-a_i)$		
3	$\max_{1\le j\le 4}(s_{j+3}-s_j)$		$\min_{1\le i\le 4}(a_{i+2}-a_i)$		
4	$\max_{1\le j\le 4}(s_{j+4}-s_j)$		$\min_{1\le i\le 4}(a_{i+3}-a_i)$		

6-2= 4	0	4
7-2= 5	6-5 =1	4
11-2 =9	6-3 =3	6
12-2 =10	6-0 =6	4

(5)) Please compute the worst-case response time



Answer = 9