

2、

(1)

signal: 0 response time: 1.44

signal: 1 response time: 2.04

signal: 2 response time: 2.56

signal: 3 response time: 3.16

signal: 4 response time: 3.68

signal: 5 response time: 4.28

signal: 6 response time: 5.2

signal: 7 response time: 8.4

signal: 8 response time: 9.0

signal: 9 response time: 9.68

signal: 10 response time: 10.2

signal: 11 response time: 19.36

signal: 12 response time: 19.8

signal: 13 response time: 20.32

signal: 14 response time: 29.400000000000002

signal: 15 response time: 29.759999999999998

signal: 16 response time: 30.279999999999998

(2)

import numpy as np

import math

signal\_num = 0

one\_bit\_trans = 0

total\_qi = 0

for idx, line in enumerate(open("input.dat", 'r')):

    item = line.rstrip()

    split\_item = item.split()

    if idx == 0:

        signal\_num = int(split\_item[0])

        trans\_time = np.zeros(signal\_num)

        period\_time = np.zeros(signal\_num)

    elif idx == 1:

        one\_bit\_trans = float(split\_item[0])

    else:

        trans\_time[int(split\_item[0])] = float(split\_item[1])

        period\_time[int(split\_item[0])] = float(split\_item[2])

for i in range(signal\_num):

    block\_time = np.max(trans\_time[i:])

    high\_priority\_signal = trans\_time[:i]

    LHS = block\_time

    while 1:

        RHS = block\_time

        for j in range(len(high\_priority\_signal)):

            RHS += math.ceil((one\_bit\_trans + LHS)/period\_time[j])\*high\_priority\_signal[j]

        if RHS == LHS:

            print("signal: %s response time: %s"%(i, (RHS + trans\_time[i])))

            break

        elif RHS >= LHS:

            LHS = RHS

        else:

            print("error in message %s"% (i))

            break