1. **Обчисліть час відповіді та час виконання під час виконання трьох завдань довжиною 200 за допомогою планувальників SJF і FIFO.**

$ ./scheduler.py -p SJF -l 200,200,200 -c

response time: 0, 200, 400

turnaround time: 200, 400, 600

$ ./scheduler.py -p FIFO -l 200,200,200 -c

response time: 0, 200, 400

turnaround time: 200, 400, 600

1. **Тепер виконайте те ж саме, але з завданнями різної довжини: 100, 200 і 300.**

$ ./scheduler.py -p SJF -l 100,200,300 -c

response time: 0, 100, 300

turnaround time: 100, 300, 600

$ ./scheduler.py -p FIFO -l 100,200,300 -c

response time: 0, 100, 300

turnaround time: 100, 300, 600

1. **Тепер зробіть те саме, але також із планувальником RR і часовим відрізком 1.**

$ ./scheduler.py -p RR -q 1 -l 100,200,300 -c

response time: 0, 1, 2

turnaround time: 298, 499, 600

1. **Для яких типів робочих навантажень SJF забезпечує той самий час виконання, що й FIFO?**

Роботи розташовані в порядку зростання довжини.

1. **Для яких типів робочих навантажень і квантових довжин SJF забезпечує такий самий час відгуку, як і RR?**

Довжина завдань однакова, а квантова довжина дорівнює довжині завдання.

1. **Що відбувається з часом відповіді з SJF, коли тривалість роботи збільшується? Чи можете ви використати симулятор, щоб продемонструвати тенденцію?**

Час відповіді збільшиться.

$ ./scheduler.py -p SJF -l 200,200,200 -c

$ ./scheduler.py -p SJF -l 300,300,300 -c

$ ./scheduler.py -p SJF -l 400,400,400 -c

1. **Що відбувається з часом відгуку з RR, коли квантові довжини збільшуються? Чи можете ви написати рівняння, яке дає найгірший час відповіді за N завдань?**

1 <= n <= N, q = квантовий час, jt = час виконання завдання

Час відгуку збільшиться, коли q < jt:

Час відповіді n-го завдання = (n - 1) \* q

Середній час відповіді = (N - 1) \* q / 2

Коли q >= jt:

Час відповіді n-го завдання = (n - 1) \* jt

Середній час відповіді = (N - 1) \* jt / 2