Bil 461 Hw – 2

Fatih Furkan HAS
141101024

1-) Burada FIFO kulanıldığı için aşağıdaki gibi bi çizelge çizerek çalışma zamanını hesaplayabiliriz.

---

0 1 2 3 4 5 6 7 8 9 10 11 15
Job-0 1 iş bitti 2. iş bitti
0. iş bitti Job-2
Job-1

Görüldüğü gibi 15 saniye sonunda tüm işler biter.

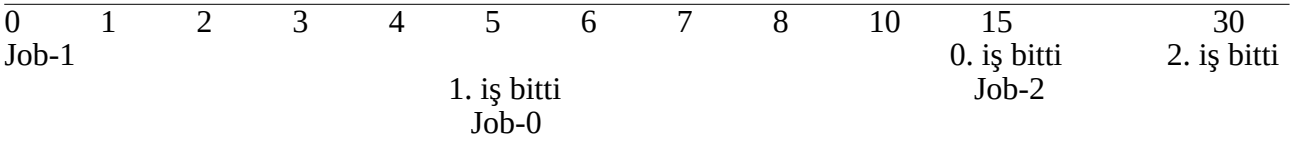
0. iş hiç beklemedi, wait = 0, ve 2. saniyede bitti.
1. iş 2 saniye bekledi, wait = 2, ve 7. saniyede bitti.
2. iş 7 saniye bekledi, wait = 7, ve 15. saniyede bitti.

Ortalama response = (0 + 2 + 7) / 3 = 3
Ortalama turnaround = (15 + 7 + 2) / 3 = 8
Ortalama wait = (0 + 2 + 7) / 3 = 3

Kodu -c parametresi ile çalıştırdığımızda bu sonuçları görebiliriz.

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 100
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 100

Here is the job list, with the run time of each job:
  Job 0 ( length = 2 )
  Job 1 ( length = 5 )
  Job 2 ( length = 8 )


Compute the turnaround time, response time, and wait time for each job.
When you are done, run this program again, with the same arguments,
but with -c, which will thus provide you with the answers. You can use
-s <somenumber> or your own job list (-l 10,15,20 for example)
to generate different problems for yourself.

toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 100 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 100

Here is the job list, with the run time of each job:
  Job 0 ( length = 2 )
  Job 1 ( length = 5 )
  Job 2 ( length = 8 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 2.00 secs ( DONE at 2.00 )
  [ time   2 ] Run job 1 for 5.00 secs ( DONE at 7.00 )
  [ time   7 ] Run job 2 for 8.00 secs ( DONE at 15.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 2.00  Wait 0.00
  Job   1 -- Response: 2.00  Turnaround 7.00  Wait 2.00
  Job   2 -- Response: 7.00  Turnaround 15.00  Wait 7.00

  Average -- Response: 3.00  Turnaround 8.00  Wait 3.00
```

Readme dosyasında belirtildiği gibi SFJ kodunu calıştırırsak burada Shortest Job First (SFJ) kullanilacağı için sonucun aşağıdaki gibi olmasını bekledim. (Burada Readme dosyasında belirtildiğinden biraz farklı şekilde çalıştırdım kodu en kısa işi Job-2 olarak tanımladım, ./scheduler.py -p SJF -l 10,5,15 şeklinde calıştırdım)



0. iş 5 saniye bekledi, wait = 5, ve 15. saniyede bitti.
1. hiç beklemedi, wait = 0, ve 5. saniyede bitti.
2. iş 15 saniye bekledi, wait = 15, ve 30. saniyede bitti.

Ortalama response = (5 + 0 + 15) / 3 = 20/3 = 6.67
Ortalama turnaround = (15 + 5 + 30) / 3 = 50/3 = 16.67
Ortalama wait = (5 + 0 + 15) / 3 = 20/3 = 6.67

Kodu -c parametresi ile çalıştırdığımızda bu sonuçları görebiliriz.

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p SJF -l 10,5,15
ARG policy SJF
ARG jlist 10,5,15

Here is the job list, with the run time of each job:
  Job 0 ( length = 10.0 )
  Job 1 ( length = 5.0 )
  Job 2 ( length = 15.0 )


Compute the turnaround time, response time, and wait time for each job.
When you are done, run this program again, with the same arguments,
but with -c, which will thus provide you with the answers. You can use
-s <somenumber> or your own job list (-l 10,15,20 for example)
to generate different problems for yourself.

toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p SJF -l 10,5,15 -c
ARG policy SJF
ARG jlist 10,5,15

Here is the job list, with the run time of each job:
  Job 0 ( length = 10.0 )
  Job 1 ( length = 5.0 )
  Job 2 ( length = 15.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 1 for 5.00 secs ( DONE at 5.00 )
  [ time   5 ] Run job 0 for 10.00 secs ( DONE at 15.00 )
  [ time  15 ] Run job 2 for 15.00 secs ( DONE at 30.00 )

Final statistics:
  Job   1 -- Response: 0.00  Turnaround 5.00  Wait 0.00
  Job   0 -- Response: 5.00  Turnaround 15.00  Wait 5.00
  Job   2 -- Response: 15.00  Turnaround 30.00  Wait 15.00

  Average -- Response: 6.67  Turnaround 16.67  Wait 6.67
```

Eğer ilk kısımda FIFO ile çalıştırdığımız işleri SFJ ile çalıştırırsak da aynı süreleri elde ederiz.

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p SJF -l 2,5,8 -c
ARG policy SJF
ARG jlist 2,5,8

Here is the job list, with the run time of each job:
  Job 0 ( length = 2.0 )
  Job 1 ( length = 5.0 )
  Job 2 ( length = 8.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 2.00 secs ( DONE at 2.00 )
  [ time   2 ] Run job 1 for 5.00 secs ( DONE at 7.00 )
  [ time   7 ] Run job 2 for 8.00 secs ( DONE at 15.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 2.00  Wait 0.00
  Job   1 -- Response: 2.00  Turnaround 7.00  Wait 2.00
  Job   2 -- Response: 7.00  Turnaround 15.00  Wait 7.00

  Average -- Response: 3.00  Turnaround 8.00  Wait 3.00

toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 100 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 100

Here is the job list, with the run time of each job:
  Job 0 ( length = 2 )
  Job 1 ( length = 5 )
  Job 2 ( length = 8 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 2.00 secs ( DONE at 2.00 )
  [ time   2 ] Run job 1 for 5.00 secs ( DONE at 7.00 )
  [ time   7 ] Run job 2 for 8.00 secs ( DONE at 15.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 2.00  Wait 0.00
  Job   1 -- Response: 2.00  Turnaround 7.00  Wait 2.00
  Job   2 -- Response: 7.00  Turnaround 15.00  Wait 7.00

  Average -- Response: 3.00  Turnaround 8.00  Wait 3.00
```

!!!İLK SORUNUN 4 İŞLEM İÇİN YAPILACAĞINI SONRADAN FARK ETTİM 4 İŞLEM İÇİN DE ÇALIŞTIRDIM ANCAK ÇIKTILARIMI DEĞİŞTİRMEDİM!!!

2-) Burada da aynı programı farklı uzunluklarda programlarla çalıştırdım yine aynı şekilde sonuçları gözlemledim. Örnek çıktılar aşağıdadır.

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 100 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 100

Here is the job list, with the run time of each job:
  Job 0 ( length = 2 )
  Job 1 ( length = 5 )
  Job 2 ( length = 8 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 2.00 secs ( DONE at 2.00 )
  [ time   2 ] Run job 1 for 5.00 secs ( DONE at 7.00 )
  [ time   7 ] Run job 2 for 8.00 secs ( DONE at 15.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 2.00  Wait 0.00
  Job   1 -- Response: 2.00  Turnaround 7.00  Wait 2.00
  Job   2 -- Response: 7.00  Turnaround 15.00  Wait 7.00

  Average -- Response: 3.00  Turnaround 8.00  Wait 3.00

toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 200 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 200

Here is the job list, with the run time of each job:
  Job 0 ( length = 1 )
  Job 1 ( length = 3 )
  Job 2 ( length = 8 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 1.00 secs ( DONE at 1.00 )
  [ time   1 ] Run job 1 for 3.00 secs ( DONE at 4.00 )
  [ time   4 ] Run job 2 for 8.00 secs ( DONE at 12.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 1.00  Wait 0.00
  Job   1 -- Response: 1.00  Turnaround 4.00  Wait 1.00
  Job   2 -- Response: 4.00  Turnaround 12.00  Wait 4.00

  Average -- Response: 1.67  Turnaround 5.67  Wait 1.67
```

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 300 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 300

Here is the job list, with the run time of each job:
  Job 0 ( length = 6 )
  Job 1 ( length = 4 )
  Job 2 ( length = 1 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 6.00 secs ( DONE at 6.00 )
  [ time   6 ] Run job 1 for 4.00 secs ( DONE at 10.00 )
  [ time  10 ] Run job 2 for 1.00 secs ( DONE at 11.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 6.00  Wait 0.00
  Job   1 -- Response: 6.00  Turnaround 10.00  Wait 6.00
  Job   2 -- Response: 10.00  Turnaround 11.00  Wait 10.00

  Average -- Response: 5.33  Turnaround 9.00  Wait 5.33

toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 400 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 400

Here is the job list, with the run time of each job:
  Job 0 ( length = 4 )
  Job 1 ( length = 3 )
  Job 2 ( length = 9 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 4.00 secs ( DONE at 4.00 )
  [ time   4 ] Run job 1 for 3.00 secs ( DONE at 7.00 )
  [ time   7 ] Run job 2 for 9.00 secs ( DONE at 16.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 4.00  Wait 0.00
  Job   1 -- Response: 4.00  Turnaround 7.00  Wait 4.00
  Job   2 -- Response: 7.00  Turnaround 16.00  Wait 7.00

  Average -- Response: 3.67  Turnaround 9.00  Wait 3.67
```

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 3 -s 500 -c
ARG policy FIFO
ARG jobs 3
ARG maxlen 10
ARG seed 500

Here is the job list, with the run time of each job:
  Job 0 ( length = 8 )
  Job 1 ( length = 10 )
  Job 2 ( length = 6 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 8.00 secs ( DONE at 8.00 )
  [ time   8 ] Run job 1 for 10.00 secs ( DONE at 18.00 )
  [ time  18 ] Run job 2 for 6.00 secs ( DONE at 24.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 8.00  Wait 0.00
  Job   1 -- Response: 8.00  Turnaround 18.00  Wait 8.00
  Job   2 -- Response: 18.00  Turnaround 24.00  Wait 18.00

  Average -- Response: 8.67  Turnaround 16.67  Wait 8.67
```

3-) RR scheduler kullandığımızda FIFO'ya ve RSJ'ye göre response time ortalamamız çok daha düşük oldu çünkü işlemler bitse de bitmese 3 saniye içinde diğer işleme geçildiği için işlemlerin ilk çalışma zamanları kısaldı. Ancak ortalama olarak turnaround ve wait sürelerimiz arttı. Aşağıda bu karşılaştırmayı yaparken gördüğüm ekran çıktıları mevcut.

FIFO:

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p FIFO -j 4 -s 500 -c
ARG policy FIFO
ARG jobs 4
ARG maxlen 10
ARG seed 500

Here is the job list, with the run time of each job:
  Job 0 ( length = 8 )
  Job 1 ( length = 10 )
  Job 2 ( length = 6 )
  Job 3 ( length = 3 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 8.00 secs ( DONE at 8.00 )
  [ time   8 ] Run job 1 for 10.00 secs ( DONE at 18.00 )
  [ time  18 ] Run job 2 for 6.00 secs ( DONE at 24.00 )
  [ time  24 ] Run job 3 for 3.00 secs ( DONE at 27.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 8.00  Wait 0.00
  Job   1 -- Response: 8.00  Turnaround 18.00  Wait 8.00
  Job   2 -- Response: 18.00  Turnaround 24.00  Wait 18.00
  Job   3 -- Response: 24.00  Turnaround 27.00  Wait 24.00

  Average -- Response: 12.50  Turnaround 19.25  Wait 12.50
```

SJF:

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p SJF -l 8,10,6,3 -c
ARG policy SJF
ARG jlist 8,10,6,3

Here is the job list, with the run time of each job:
  Job 0 ( length = 8.0 )
  Job 1 ( length = 10.0 )
  Job 2 ( length = 6.0 )
  Job 3 ( length = 3.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 3 for 3.00 secs ( DONE at 3.00 )
  [ time   3 ] Run job 2 for 6.00 secs ( DONE at 9.00 )
  [ time   9 ] Run job 0 for 8.00 secs ( DONE at 17.00 )
  [ time  17 ] Run job 1 for 10.00 secs ( DONE at 27.00 )

Final statistics:
  Job   3 -- Response: 0.00  Turnaround 3.00  Wait 0.00
  Job   2 -- Response: 3.00  Turnaround 9.00  Wait 3.00
  Job   0 -- Response: 9.00  Turnaround 17.00  Wait 9.00
  Job   1 -- Response: 17.00  Turnaround 27.00  Wait 17.00

  Average -- Response: 7.25  Turnaround 14.00  Wait 7.25
```

RR:

```
toor@001:~/Desktop/HW-Scheduler$ ./scheduler.py -p RR -j 4 -s 500 -q 3 -c
ARG policy RR
ARG jobs 4
ARG maxlen 10
ARG seed 500

Here is the job list, with the run time of each job:
  Job 0 ( length = 8 )
  Job 1 ( length = 10 )
  Job 2 ( length = 6 )
  Job 3 ( length = 3 )


** Solutions **

Execution trace:
  [ time   0 ] Run job   0 for 3.00 secs
  [ time   3 ] Run job   1 for 3.00 secs
  [ time   6 ] Run job   2 for 3.00 secs
  [ time   9 ] Run job   3 for 3.00 secs ( DONE at 12.00 )
  [ time  12 ] Run job   0 for 3.00 secs
  [ time  15 ] Run job   1 for 3.00 secs
  [ time  18 ] Run job   2 for 3.00 secs ( DONE at 21.00 )
  [ time  21 ] Run job   0 for 2.00 secs ( DONE at 23.00 )
  [ time  23 ] Run job   1 for 3.00 secs
  [ time  26 ] Run job   1 for 1.00 secs ( DONE at 27.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 23.00  Wait 15.00
  Job   1 -- Response: 3.00  Turnaround 27.00  Wait 17.00
  Job   2 -- Response: 6.00  Turnaround 21.00  Wait 15.00
  Job   3 -- Response: 9.00  Turnaround 12.00  Wait 9.00

  Average -- Response: 4.50  Turnaround 20.75  Wait 14.00
```

4-) Eğer STCF kullanmış olsaydık FIFO'ya ve SJF'ye göre ortalama response time düşerdi. STCF response time'i minimize etmek için kullanılabilir. Turnaround time için konuşacak olursak STCF sayesinde turnaround time'da düşürülür.

Part 2

1-)

```
toor@001:~/Desktop/HW-MLFQ$ ./mlfq.py -j 2 -i 0 -m 50 -n 2 -c
Here is the list of inputs:
OPTIONS jobs 2
OPTIONS queues 2
OPTIONS quantum length for queue  1 is  10
OPTIONS quantum length for queue  0 is  10
OPTIONS boost 0
OPTIONS ioTime 0
OPTIONS stayAfterIO False
OPTIONS iobump False


For each job, three defining characteristics are given:
  startTime : at what time does the job enter the system
  runTime   : the total CPU time needed by the job to finish
  ioFreq    : every ioFreq time units, the job issues an I/O
              (the I/O takes ioTime units to complete)

Job List:
  Job  0: startTime   0 - runTime  42 - ioFreq   7
  Job  1: startTime   0 - runTime  21 - ioFreq   3


Execution Trace:

[ time 0 ] JOB BEGINS by JOB 0
[ time 0 ] JOB BEGINS by JOB 1
[ time 0 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 9 RUNTIME 42 TIMELEFT 41 ]
[ time 1 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 8 RUNTIME 42 TIMELEFT 40 ]
[ time 2 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 7 RUNTIME 42 TIMELEFT 39 ]
[ time 3 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 6 RUNTIME 42 TIMELEFT 38 ]
[ time 4 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 5 RUNTIME 42 TIMELEFT 37 ]
[ time 5 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 4 RUNTIME 42 TIMELEFT 36 ]
[ time 6 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 3 RUNTIME 42 TIMELEFT 35 ]
[ time 7 ] IO_START by JOB 0
IO DONE
[ time 7 ] IO_DONE by JOB 0
[ time 7 ] Run JOB 1 at PRIORITY 1 [ TICKSLEFT 9 RUNTIME 21 TIMELEFT 20 ]
[ time 8 ] Run JOB 1 at PRIORITY 1 [ TICKSLEFT 8 RUNTIME 21 TIMELEFT 19 ]
[ time 9 ] Run JOB 1 at PRIORITY 1 [ TICKSLEFT 7 RUNTIME 21 TIMELEFT 18 ]
[ time 10 ] IO_START by JOB 1
IO DONE
[ time 10 ] IO_DONE by JOB 1
[ time 10 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 2 RUNTIME 42 TIMELEFT 34 ]
[ time 11 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 1 RUNTIME 42 TIMELEFT 33 ]
[ time 12 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 0 RUNTIME 42 TIMELEFT 32 ]
[ time 13 ] Run JOB 1 at PRIORITY 1 [ TICKSLEFT 6 RUNTIME 21 TIMELEFT 17 ]
[ time 14 ] Run JOB 1 at PRIORITY 1 [ TICKSLEFT 5 RUNTIME 21 TIMELEFT 16 ]
[ time 15 ] Run JOB 1 at PRIORITY 1 [ TICKSLEFT 4 RUNTIME 21 TIMELEFT 15 ]
[ time 16 ] IO_START by JOB 1
```

IO DONE
[time 16] IO_DONE by JOB 1
[time 16] Run JOB 1 at PRIORITY 1 [TICKSLEFT 3 RUNTIME 21 TIMELEFT 14]
[time 17] Run JOB 1 at PRIORITY 1 [TICKSLEFT 2 RUNTIME 21 TIMELEFT 13]
[time 18] Run JOB 1 at PRIORITY 1 [TICKSLEFT 1 RUNTIME 21 TIMELEFT 12]
[time 19] IO_START by JOB 1
IO DONE
[time 19] IO_DONE by JOB 1
[time 19] Run JOB 1 at PRIORITY 1 [TICKSLEFT 0 RUNTIME 21 TIMELEFT 11]
[time 20] Run JOB 0 at PRIORITY 0 [TICKSLEFT 9 RUNTIME 42 TIMELEFT 31]
[time 21] Run JOB 0 at PRIORITY 0 [TICKSLEFT 8 RUNTIME 42 TIMELEFT 30]
[time 22] Run JOB 0 at PRIORITY 0 [TICKSLEFT 7 RUNTIME 42 TIMELEFT 29]
[time 23] Run JOB 0 at PRIORITY 0 [TICKSLEFT 6 RUNTIME 42 TIMELEFT 28]
[time 24] IO_START by JOB 0
IO DONE
[time 24] IO_DONE by JOB 0
[time 24] Run JOB 1 at PRIORITY 0 [TICKSLEFT 9 RUNTIME 21 TIMELEFT 10]
[time 25] Run JOB 1 at PRIORITY 0 [TICKSLEFT 8 RUNTIME 21 TIMELEFT 9]
[time 26] IO_START by JOB 1
IO DONE
[time 26] IO_DONE by JOB 1
[time 26] Run JOB 0 at PRIORITY 0 [TICKSLEFT 5 RUNTIME 42 TIMELEFT 27]
[time 27] Run JOB 0 at PRIORITY 0 [TICKSLEFT 4 RUNTIME 42 TIMELEFT 26]
[time 28] Run JOB 0 at PRIORITY 0 [TICKSLEFT 3 RUNTIME 42 TIMELEFT 25]
[time 29] Run JOB 0 at PRIORITY 0 [TICKSLEFT 2 RUNTIME 42 TIMELEFT 24]
[time 30] Run JOB 0 at PRIORITY 0 [TICKSLEFT 1 RUNTIME 42 TIMELEFT 23]
[time 31] Run JOB 0 at PRIORITY 0 [TICKSLEFT 0 RUNTIME 42 TIMELEFT 22]
[time 32] Run JOB 1 at PRIORITY 0 [TICKSLEFT 7 RUNTIME 21 TIMELEFT 8]
[time 33] Run JOB 1 at PRIORITY 0 [TICKSLEFT 6 RUNTIME 21 TIMELEFT 7]
[time 34] Run JOB 1 at PRIORITY 0 [TICKSLEFT 5 RUNTIME 21 TIMELEFT 6]
[time 35] IO_START by JOB 1
IO DONE
[time 35] IO_DONE by JOB 1
[time 35] Run JOB 0 at PRIORITY 0 [TICKSLEFT 9 RUNTIME 42 TIMELEFT 21]
[time 36] IO_START by JOB 0
IO DONE
[time 36] IO_DONE by JOB 0
[time 36] Run JOB 1 at PRIORITY 0 [TICKSLEFT 4 RUNTIME 21 TIMELEFT 5]
[time 37] Run JOB 1 at PRIORITY 0 [TICKSLEFT 3 RUNTIME 21 TIMELEFT 4]
[time 38] Run JOB 1 at PRIORITY 0 [TICKSLEFT 2 RUNTIME 21 TIMELEFT 3]
[time 39] IO_START by JOB 1
IO DONE
[time 39] IO_DONE by JOB 1
[time 39] Run JOB 0 at PRIORITY 0 [TICKSLEFT 8 RUNTIME 42 TIMELEFT 20]
[time 40] Run JOB 0 at PRIORITY 0 [TICKSLEFT 7 RUNTIME 42 TIMELEFT 19]
[time 41] Run JOB 0 at PRIORITY 0 [TICKSLEFT 6 RUNTIME 42 TIMELEFT 18]
[time 42] Run JOB 0 at PRIORITY 0 [TICKSLEFT 5 RUNTIME 42 TIMELEFT 17]
[time 43] Run JOB 0 at PRIORITY 0 [TICKSLEFT 4 RUNTIME 42 TIMELEFT 16]
[time 44] Run JOB 0 at PRIORITY 0 [TICKSLEFT 3 RUNTIME 42 TIMELEFT 15]
[time 45] Run JOB 0 at PRIORITY 0 [TICKSLEFT 2 RUNTIME 42 TIMELEFT 14]
[time 46] IO_START by JOB 0
IO DONE
[time 46] IO_DONE by JOB 0
[time 46] Run JOB 1 at PRIORITY 0 [TICKSLEFT 1 RUNTIME 21 TIMELEFT 2]
[time 47] Run JOB 1 at PRIORITY 0 [TICKSLEFT 0 RUNTIME 21 TIMELEFT 1]
[time 48] Run JOB 0 at PRIORITY 0 [TICKSLEFT 1 RUNTIME 42 TIMELEFT 13]
[time 49] Run JOB 0 at PRIORITY 0 [TICKSLEFT 0 RUNTIME 42 TIMELEFT 12]

[time 50] Run JOB 1 at PRIORITY 0 [TICKSLEFT 9 RUNTIME 21 TIMELEFT 0]
[time 51] FINISHED JOB 1
[time 51] Run JOB 0 at PRIORITY 0 [TICKSLEFT 9 RUNTIME 42 TIMELEFT 11]
[time 52] Run JOB 0 at PRIORITY 0 [TICKSLEFT 8 RUNTIME 42 TIMELEFT 10]
[time 53] Run JOB 0 at PRIORITY 0 [TICKSLEFT 7 RUNTIME 42 TIMELEFT 9]
[time 54] Run JOB 0 at PRIORITY 0 [TICKSLEFT 6 RUNTIME 42 TIMELEFT 8]
[time 55] Run JOB 0 at PRIORITY 0 [TICKSLEFT 5 RUNTIME 42 TIMELEFT 7]
[time 56] IO_START by JOB 0
IO DONE
[time 56] IO_DONE by JOB 0
[time 56] Run JOB 0 at PRIORITY 0 [TICKSLEFT 4 RUNTIME 42 TIMELEFT 6]
[time 57] Run JOB 0 at PRIORITY 0 [TICKSLEFT 3 RUNTIME 42 TIMELEFT 5]
[time 58] Run JOB 0 at PRIORITY 0 [TICKSLEFT 2 RUNTIME 42 TIMELEFT 4]
[time 59] Run JOB 0 at PRIORITY 0 [TICKSLEFT 1 RUNTIME 42 TIMELEFT 3]
[time 60] Run JOB 0 at PRIORITY 0 [TICKSLEFT 0 RUNTIME 42 TIMELEFT 2]
[time 61] Run JOB 0 at PRIORITY 0 [TICKSLEFT 9 RUNTIME 42 TIMELEFT 1]
[time 62] Run JOB 0 at PRIORITY 0 [TICKSLEFT 8 RUNTIME 42 TIMELEFT 0]
[time 63] FINISHED JOB 0

Final statistics:
  Job 0: startTime   0 - response   0 - turnaround 63
  Job 1: startTime   0 - response   7 - turnaround 51

  Avg 1: startTime n/a - response 3.50 - turnaround 57.00

2 job ve 2 queu için I/O time’i 0 yaptım ve max length’i 50’ye ayarlayarak çalıştırdım çıktı yukarıdaki gibi oluştu.

2-)

```
toor@001:~/Desktop/HW-MLFQ$ ./mlfq.py --jlist 0,20,0:0,50,8 -S -q 10 -i 2 -
Here is the list of inputs:
OPTIONS jobs 2
OPTIONS queues 3
OPTIONS quantum length for queue  2 is  10
OPTIONS quantum length for queue  1 is  10
OPTIONS quantum length for queue  0 is  10
OPTIONS boost 0
OPTIONS ioTime 2
OPTIONS stayAfterIO True
OPTIONS iobump False


For each job, three defining characteristics are given:
  startTime : at what time does the job enter the system
  runTime   : the total CPU time needed by the job to finish
  ioFreq    : every ioFreq time units, the job issues an I/O
              (the I/O takes ioTime units to complete)

Job List:
  Job  0: startTime   0 - runTime  20 - ioFreq   0
  Job  1: startTime   0 - runTime  50 - ioFreq   8


Execution Trace:

[ time 0 ] JOB BEGINS by JOB 0
[ time 0 ] JOB BEGINS by JOB 1
[ time 0 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 20 TIMELEFT 19 ]
[ time 1 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 20 TIMELEFT 18 ]
[ time 2 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 20 TIMELEFT 17 ]
[ time 3 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 20 TIMELEFT 16 ]
[ time 4 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 20 TIMELEFT 15 ]
[ time 5 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 20 TIMELEFT 14 ]
[ time 6 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 20 TIMELEFT 13 ]
[ time 7 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 20 TIMELEFT 12 ]
[ time 8 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 1 RUNTIME 20 TIMELEFT 11 ]
[ time 9 ] Run JOB 0 at PRIORITY 2 [ TICKSLEFT 0 RUNTIME 20 TIMELEFT 10 ]
[ time 10 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 49 ]
[ time 11 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 48 ]
[ time 12 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 50 TIMELEFT 47 ]
[ time 13 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 50 TIMELEFT 46 ]
[ time 14 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 50 TIMELEFT 45 ]
[ time 15 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 50 TIMELEFT 44 ]
[ time 16 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 50 TIMELEFT 43 ]
[ time 17 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 50 TIMELEFT 42 ]
[ time 18 ] IO_START by JOB 1
IO DONE
[ time 18 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 9 RUNTIME 20 TIMELEFT 9 ]
[ time 19 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 8 RUNTIME 20 TIMELEFT 8 ]
[ time 20 ] IO_DONE by JOB 1
[ time 20 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 41 ]
[ time 21 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 40 ]
[ time 22 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 50 TIMELEFT 39 ]
[ time 23 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 50 TIMELEFT 38 ]
[ time 24 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 50 TIMELEFT 37 ]
[ time 25 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 50 TIMELEFT 36 ]
[ time 26 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 50 TIMELEFT 35 ]
[ time 27 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 50 TIMELEFT 34 ]
[ time 28 ] IO_START by JOB 1
IO DONE
[ time 28 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 7 RUNTIME 20 TIMELEFT 7 ]
[ time 29 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 6 RUNTIME 20 TIMELEFT 6 ]
[ time 30 ] IO_DONE by JOB 1
[ time 30 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 33 ]
[ time 31 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 32 ]
[ time 32 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 50 TIMELEFT 31 ]
[ time 33 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 50 TIMELEFT 30 ]
[ time 34 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 50 TIMELEFT 29 ]
[ time 35 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 50 TIMELEFT 28 ]
[ time 36 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 50 TIMELEFT 27 ]
[ time 37 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 50 TIMELEFT 26 ]
[ time 38 ] IO_START by JOB 1
IO DONE
[ time 38 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 5 RUNTIME 20 TIMELEFT 5 ]
[ time 39 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 4 RUNTIME 20 TIMELEFT 4 ]
[ time 40 ] IO_DONE by JOB 1
[ time 40 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 25 ]
[ time 41 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 24 ]
[ time 42 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 50 TIMELEFT 23 ]
[ time 43 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 50 TIMELEFT 22 ]
[ time 44 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 50 TIMELEFT 21 ]
```

```
[ time 45 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 50 TIMELEFT 20 ]
[ time 46 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 50 TIMELEFT 19 ]
[ time 47 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 50 TIMELEFT 18 ]
[ time 48 ] IO_START by JOB 1
IO DONE
[ time 48 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 3 RUNTIME 20 TIMELEFT 3 ]
[ time 49 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 2 RUNTIME 20 TIMELEFT 2 ]
[ time 50 ] IO_DONE by JOB 1
[ time 50 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 17 ]
[ time 51 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 16 ]
[ time 52 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 50 TIMELEFT 15 ]
[ time 53 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 50 TIMELEFT 14 ]
[ time 54 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 50 TIMELEFT 13 ]
[ time 55 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 50 TIMELEFT 12 ]
[ time 56 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 50 TIMELEFT 11 ]
[ time 57 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 50 TIMELEFT 10 ]
[ time 58 ] IO_START by JOB 1
IO DONE
[ time 58 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 1 RUNTIME 20 TIMELEFT 1 ]
[ time 59 ] Run JOB 0 at PRIORITY 1 [ TICKSLEFT 0 RUNTIME 20 TIMELEFT 0 ]
[ time 60 ] FINISHED JOB 0
[ time 60 ] IO_DONE by JOB 1
[ time 60 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 9 ]
[ time 61 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 8 ]
[ time 62 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 7 RUNTIME 50 TIMELEFT 7 ]
[ time 63 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 6 RUNTIME 50 TIMELEFT 6 ]
[ time 64 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 5 RUNTIME 50 TIMELEFT 5 ]
[ time 65 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 4 RUNTIME 50 TIMELEFT 4 ]
[ time 66 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 3 RUNTIME 50 TIMELEFT 3 ]
[ time 67 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 2 RUNTIME 50 TIMELEFT 2 ]
[ time 68 ] IO_START by JOB 1
IO DONE
[ time 68 ] IDLE
[ time 69 ] IDLE
[ time 70 ] IO_DONE by JOB 1
[ time 70 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 9 RUNTIME 50 TIMELEFT 1 ]
[ time 71 ] Run JOB 1 at PRIORITY 2 [ TICKSLEFT 8 RUNTIME 50 TIMELEFT 0 ]
[ time 72 ] FINISHED JOB 1

Final statistics:
  Job  0: startTime   0 - response   0 - turnaround  60
  Job  1: startTime   0 - response  10 - turnaround  72

  Avg  1: startTime n/a - response 5.00 - turnaround 66.00
```

./mlfq.py --jlist 0,20,0:0,50,8 -S -q 10 -i 2 -c kodumuzu bu şekilde çalıştırırsak eğer sistemi kandırarak priortylere göre diğer işlemlere geçmesini engelleyebiliriz. Burada yapmaya çalıştığımız şey ikinci işlem çalışırken ona bir I/O işlemi verip priorty'sinin düşmesini engelledik bu sayede sadece job 1'in I/O işlemleri sırasında job 0 çalışabildi.

Part – 3

1-)

```
toor@001:~/Desktop/HW-Lottery$ ./lottery.py -j 4 -s 1 -c
ARG jlist
ARG jobs 4
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 1

Here is the job list, with the run time of each job:
  Job 0 ( length = 1, tickets = 84 )
  Job 1 ( length = 7, tickets = 25 )
  Job 2 ( length = 4, tickets = 44 )
  Job 3 ( length = 6, tickets = 78 )


** Solutions **

Random 93859 -> Winning ticket 73 (of 231) -> Run 0
  Jobs:  (* job:0 timeleft:1 tix:84 )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:4 tix:44 )   (  job:3 timeleft:6 tix:78 )
--> JOB 0 DONE at time 1
Random 28347 -> Winning ticket 123 (of 147) -> Run 3
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:4 tix:44 )   (* job:3 timeleft:6 tix:78 )
Random 835765 -> Winning ticket 70 (of 147) -> Run 3
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:4 tix:44 )   (* job:3 timeleft:5 tix:78 )
Random 432767 -> Winning ticket 146 (of 147) -> Run 3
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:4 tix:44 )   (* job:3 timeleft:4 tix:78 )
Random 762280 -> Winning ticket 85 (of 147) -> Run 3
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:4 tix:44 )   (* job:3 timeleft:3 tix:78 )
Random 2106 -> Winning ticket 48 (of 147) -> Run 2
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (* job:2 timeleft:4 tix:44 )   (  job:3 timeleft:2 tix:78 )
Random 445387 -> Winning ticket 124 (of 147) -> Run 3
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:3 tix:44 )   (* job:3 timeleft:2 tix:78 )
Random 721540 -> Winning ticket 64 (of 147) -> Run 2
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (* job:2 timeleft:3 tix:44 )   (  job:3 timeleft:1 tix:78 )
Random 228762 -> Winning ticket 30 (of 147) -> Run 2
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (* job:2 timeleft:2 tix:44 )   (  job:3 timeleft:1 tix:78 )
Random 945271 -> Winning ticket 61 (of 147) -> Run 2
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (* job:2 timeleft:1 tix:44 )   (  job:3 timeleft:1 tix:78 )
--> JOB 2 DONE at time 10
Random 901428 -> Winning ticket 75 (of 103) -> Run 3
  Jobs:  (  job:0 timeleft:0 tix:--- )   (  job:1 timeleft:7 tix:25 )   (  job:2 timeleft:0 tix:--- )   (* job:3 timeleft:1 tix:78 )
--> JOB 3 DONE at time 11
Random 30590 -> Winning ticket 15 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:7 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 25445 -> Winning ticket 20 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:6 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 541413 -> Winning ticket 13 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:5 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 939150 -> Winning ticket 0 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:4 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 381204 -> Winning ticket 4 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:3 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 216599 -> Winning ticket 24 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:2 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 422116 -> Winning ticket 16 (of 25) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:1 tix:25 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
--> JOB 1 DONE at time 18
```

```
toor@001:~/Desktop/HW-Lottery$ ./lottery.py -j 4 -s 2 -c
ARG jlist
ARG jobs 4
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 2

Here is the job list, with the run time of each job:
  Job 0 ( length = 9, tickets = 94 )
  Job 1 ( length = 8, tickets = 73 )
  Job 2 ( length = 6, tickets = 30 )
  Job 3 ( length = 6, tickets = 60 )


** Solutions **

Random 581204 -> Winning ticket 127 (of 257) -> Run 1
  Jobs:  (  job:0 timeleft:9 tix:94 )   (* job:1 timeleft:8 tix:73 )   (  job:2 timeleft:6 tix:30 )   (  job:3 timeleft:6 tix:60 )
Random 158383 -> Winning ticket 71 (of 257) -> Run 0
  Jobs:  (* job:0 timeleft:9 tix:94 )   (  job:1 timeleft:7 tix:73 )   (  job:2 timeleft:6 tix:30 )   (  job:3 timeleft:6 tix:60 )
Random 430670 -> Winning ticket 195 (of 257) -> Run 2
  Jobs:  (  job:0 timeleft:8 tix:94 )   (  job:1 timeleft:7 tix:73 )   (* job:2 timeleft:6 tix:30 )   (  job:3 timeleft:6 tix:60 )
Random 393532 -> Winning ticket 65 (of 257) -> Run 0
  Jobs:  (* job:0 timeleft:8 tix:94 )   (  job:1 timeleft:7 tix:73 )   (  job:2 timeleft:5 tix:30 )   (  job:3 timeleft:6 tix:60 )
Random 723012 -> Winning ticket 71 (of 257) -> Run 0
  Jobs:  (* job:0 timeleft:7 tix:94 )   (  job:1 timeleft:7 tix:73 )   (  job:2 timeleft:5 tix:30 )   (  job:3 timeleft:6 tix:60 )
Random 994820 -> Winning ticket 230 (of 257) -> Run 3
  Jobs:  (  job:0 timeleft:6 tix:94 )   (  job:1 timeleft:7 tix:73 )   (  job:2 timeleft:5 tix:30 )   (* job:3 timeleft:6 tix:60 )
Random 949396 -> Winning ticket 38 (of 257) -> Run 0
  Jobs:  (* job:0 timeleft:6 tix:94 )   (  job:1 timeleft:7 tix:73 )   (  job:2 timeleft:5 tix:30 )   (  job:3 timeleft:5 tix:60 )
Random 544177 -> Winning ticket 108 (of 257) -> Run 1
  Jobs:  (  job:0 timeleft:5 tix:94 )   (* job:1 timeleft:7 tix:73 )   (  job:2 timeleft:5 tix:30 )   (  job:3 timeleft:5 tix:60 )
Random 444854 -> Winning ticket 244 (of 257) -> Run 3
  Jobs:  (  job:0 timeleft:5 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:5 tix:30 )   (* job:3 timeleft:5 tix:60 )
Random 268241 -> Winning ticket 190 (of 257) -> Run 2
  Jobs:  (  job:0 timeleft:5 tix:94 )   (  job:1 timeleft:6 tix:73 )   (* job:2 timeleft:5 tix:30 )   (  job:3 timeleft:4 tix:60 )
Random 35924 -> Winning ticket 201 (of 257) -> Run 3
  Jobs:  (  job:0 timeleft:5 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:4 tix:30 )   (* job:3 timeleft:4 tix:60 )
Random 27444 -> Winning ticket 202 (of 257) -> Run 3
  Jobs:  (  job:0 timeleft:5 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:4 tix:30 )   (* job:3 timeleft:3 tix:60 )
Random 464894 -> Winning ticket 238 (of 257) -> Run 3
  Jobs:  (  job:0 timeleft:5 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:4 tix:30 )   (* job:3 timeleft:2 tix:60 )
Random 318465 -> Winning ticket 42 (of 257) -> Run 0
  Jobs:  (* job:0 timeleft:5 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:4 tix:30 )   (  job:3 timeleft:1 tix:60 )
Random 380015 -> Winning ticket 169 (of 257) -> Run 2
  Jobs:  (  job:0 timeleft:4 tix:94 )   (  job:1 timeleft:6 tix:73 )   (* job:2 timeleft:4 tix:30 )   (  job:3 timeleft:1 tix:60 )
Random 891790 -> Winning ticket 0 (of 257) -> Run 0
  Jobs:  (* job:0 timeleft:4 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:3 tix:30 )   (  job:3 timeleft:1 tix:60 )
Random 525753 -> Winning ticket 188 (of 257) -> Run 2
  Jobs:  (  job:0 timeleft:3 tix:94 )   (  job:1 timeleft:6 tix:73 )   (* job:2 timeleft:3 tix:30 )   (  job:3 timeleft:1 tix:60 )
Random 560510 -> Winning ticket 250 (of 257) -> Run 3
  Jobs:  (  job:0 timeleft:3 tix:94 )   (  job:1 timeleft:6 tix:73 )   (  job:2 timeleft:2 tix:30 )   (* job:3 timeleft:1 tix:60 )
--> JOB 3 DONE at time 18
Random 236123 -> Winning ticket 117 (of 197) -> Run 1
  Jobs:  (  job:0 timeleft:3 tix:94 )   (* job:1 timeleft:6 tix:73 )   (  job:2 timeleft:2 tix:30 )   (  job:3 timeleft:0 tix:--- )
Random 23858 -> Winning ticket 21 (of 197) -> Run 0
  Jobs:  (* job:0 timeleft:3 tix:94 )   (  job:1 timeleft:5 tix:73 )   (  job:2 timeleft:2 tix:30 )   (  job:3 timeleft:0 tix:--- )
Random 325143 -> Winning ticket 93 (of 197) -> Run 0
  Jobs:  (* job:0 timeleft:2 tix:94 )   (  job:1 timeleft:5 tix:73 )   (  job:2 timeleft:2 tix:30 )   (  job:3 timeleft:0 tix:--- )
Random 136697 -> Winning ticket 176 (of 197) -> Run 2
  Jobs:  (  job:0 timeleft:1 tix:94 )   (  job:1 timeleft:5 tix:73 )   (* job:2 timeleft:2 tix:30 )   (  job:3 timeleft:0 tix:--- )
Random 510224 -> Winning ticket 191 (of 197) -> Run 2
  Jobs:  (  job:0 timeleft:1 tix:94 )   (  job:1 timeleft:5 tix:73 )   (* job:2 timeleft:1 tix:30 )   (  job:3 timeleft:0 tix:--- )
--> JOB 2 DONE at time 23
Random 998684 -> Winning ticket 24 (of 167) -> Run 0
  Jobs:  (* job:0 timeleft:1 tix:94 )   (  job:1 timeleft:5 tix:73 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
--> JOB 0 DONE at time 24
Random 674480 -> Winning ticket 33 (of 73) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:5 tix:73 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 181843 -> Winning ticket 0 (of 73) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:4 tix:73 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 893572 -> Winning ticket 52 (of 73) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:3 tix:73 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 796760 -> Winning ticket 38 (of 73) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:2 tix:73 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
Random 734402 -> Winning ticket 22 (of 73) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:1 tix:73 )   (  job:2 timeleft:0 tix:--- )   (  job:3 timeleft:0 tix:--- )
--> JOB 1 DONE at time 29
```

2-) ./lottery.py -j 2 -s 1 -l 200:200,200:200 -c şeklinde çalıştırdığımızda böyle bir sonuç elde ediyoruz:

--> JOB 0 DONE at time 390
--> JOB 1 DONE at time 400

Burada hesaplamamızı yaparsak: 390 / 400 = 0,975 buluruz.

./lottery.py -j 2 -s 2 -l 200:200,200:200 -c şeklinde çalıştırıp seed = 2 yaparsak:

--> JOB 0 DONE at time 392
--> JOB 1 DONE at time 400

Burada hesaplamamızı yaparsak: 392 / 400 = 0,98 buluruz.

./lottery.py -j 2 -s 5 -l 200:200,200:200 -c şeklinde çalıştırıp seed = 2 yaparsak:

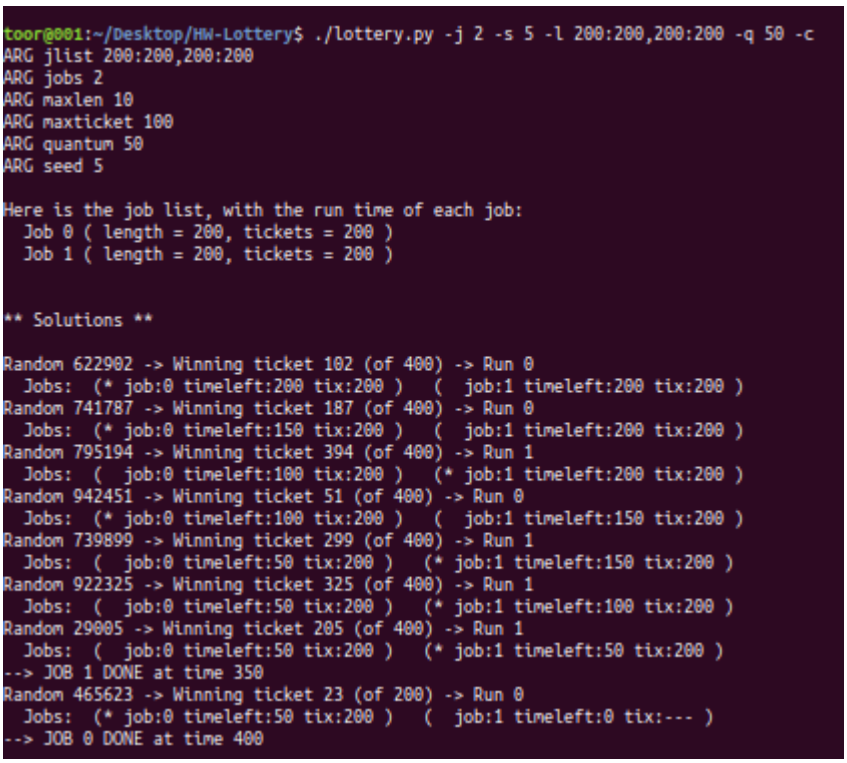
--> JOB 0 DONE at time 387
--> JOB 1 DONE at time 400

Burada hesaplamamızı yaparsak: 387 / 400 = 0,9675 buluruz.

Seed arttıkça unfair olmaktadır.

3-)

```
toor@001:~/Desktop/HW-Lottery$ ./lottery.py -j 2 -s 5 -l 200:200,200:200 -q 20 -c
ARG jlist 200:200,200:200
ARG jobs 2
ARG maxlen 10
ARG maxticket 100
ARG quantum 20
ARG seed 5

Here is the job list, with the run time of each job:
  Job 0 ( length = 200, tickets = 200 )
  Job 1 ( length = 200, tickets = 200 )


** Solutions **

Random 622902 -> Winning ticket 102 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:200 tix:200 )   (  job:1 timeleft:200 tix:200 )
Random 741787 -> Winning ticket 187 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:180 tix:200 )   (  job:1 timeleft:200 tix:200 )
Random 795194 -> Winning ticket 394 (of 400) -> Run 1
  Jobs:  (  job:0 timeleft:160 tix:200 )   (* job:1 timeleft:200 tix:200 )
Random 942451 -> Winning ticket 51 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:160 tix:200 )   (  job:1 timeleft:180 tix:200 )
Random 739899 -> Winning ticket 299 (of 400) -> Run 1
  Jobs:  (  job:0 timeleft:140 tix:200 )   (* job:1 timeleft:180 tix:200 )
Random 922325 -> Winning ticket 325 (of 400) -> Run 1
  Jobs:  (  job:0 timeleft:140 tix:200 )   (* job:1 timeleft:160 tix:200 )
Random 29005 -> Winning ticket 205 (of 400) -> Run 1
  Jobs:  (  job:0 timeleft:140 tix:200 )   (* job:1 timeleft:140 tix:200 )
Random 465623 -> Winning ticket 23 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:140 tix:200 )   (  job:1 timeleft:120 tix:200 )
Random 943357 -> Winning ticket 157 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:120 tix:200 )   (  job:1 timeleft:120 tix:200 )
Random 648975 -> Winning ticket 175 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:100 tix:200 )   (  job:1 timeleft:120 tix:200 )
Random 900901 -> Winning ticket 101 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:80 tix:200 )   (  job:1 timeleft:120 tix:200 )
Random 113206 -> Winning ticket 6 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:60 tix:200 )   (  job:1 timeleft:120 tix:200 )
Random 469069 -> Winning ticket 269 (of 400) -> Run 1
  Jobs:  (  job:0 timeleft:40 tix:200 )   (* job:1 timeleft:120 tix:200 )
Random 246573 -> Winning ticket 173 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:40 tix:200 )   (  job:1 timeleft:100 tix:200 )
Random 543761 -> Winning ticket 161 (of 400) -> Run 0
  Jobs:  (* job:0 timeleft:20 tix:200 )   (  job:1 timeleft:100 tix:200 )
--> JOB 0 DONE at time 300
Random 573941 -> Winning ticket 141 (of 200) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:100 tix:200 )
Random 13114 -> Winning ticket 114 (of 200) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:80 tix:200 )
Random 216730 -> Winning ticket 130 (of 200) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:60 tix:200 )
Random 279482 -> Winning ticket 82 (of 200) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:40 tix:200 )
Random 916346 -> Winning ticket 146 (of 200) -> Run 1
  Jobs:  (  job:0 timeleft:0 tix:--- )   (* job:1 timeleft:20 tix:200 )
--> JOB 1 DONE at time 400
```



Görüldüğü gibi q değerimizi arttırdığımızda ilk işlem daha kısa sürede tamamlanmakta ve unfairness artmaktadır.