Jayven Larsen

101260364

Aj Donald

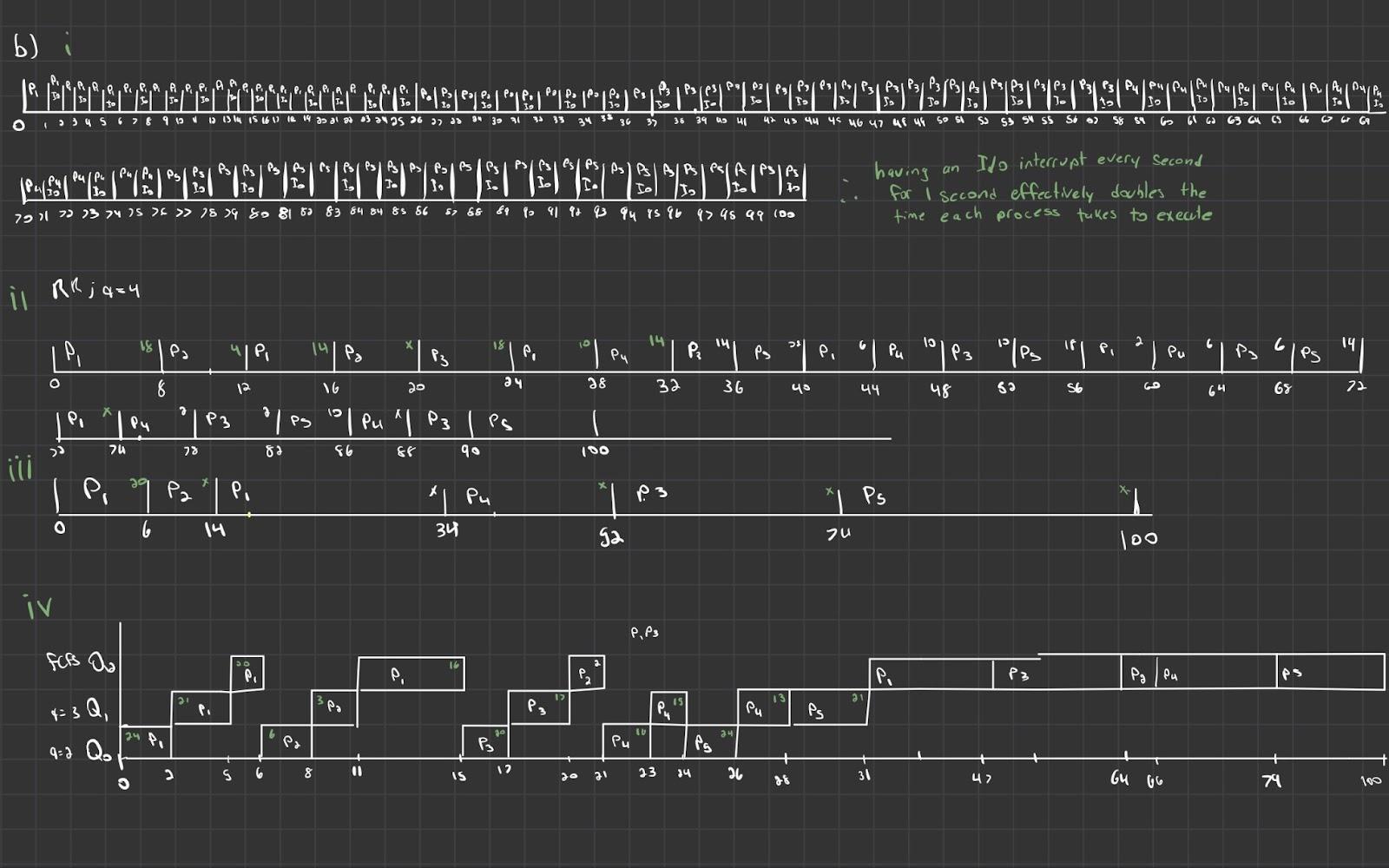
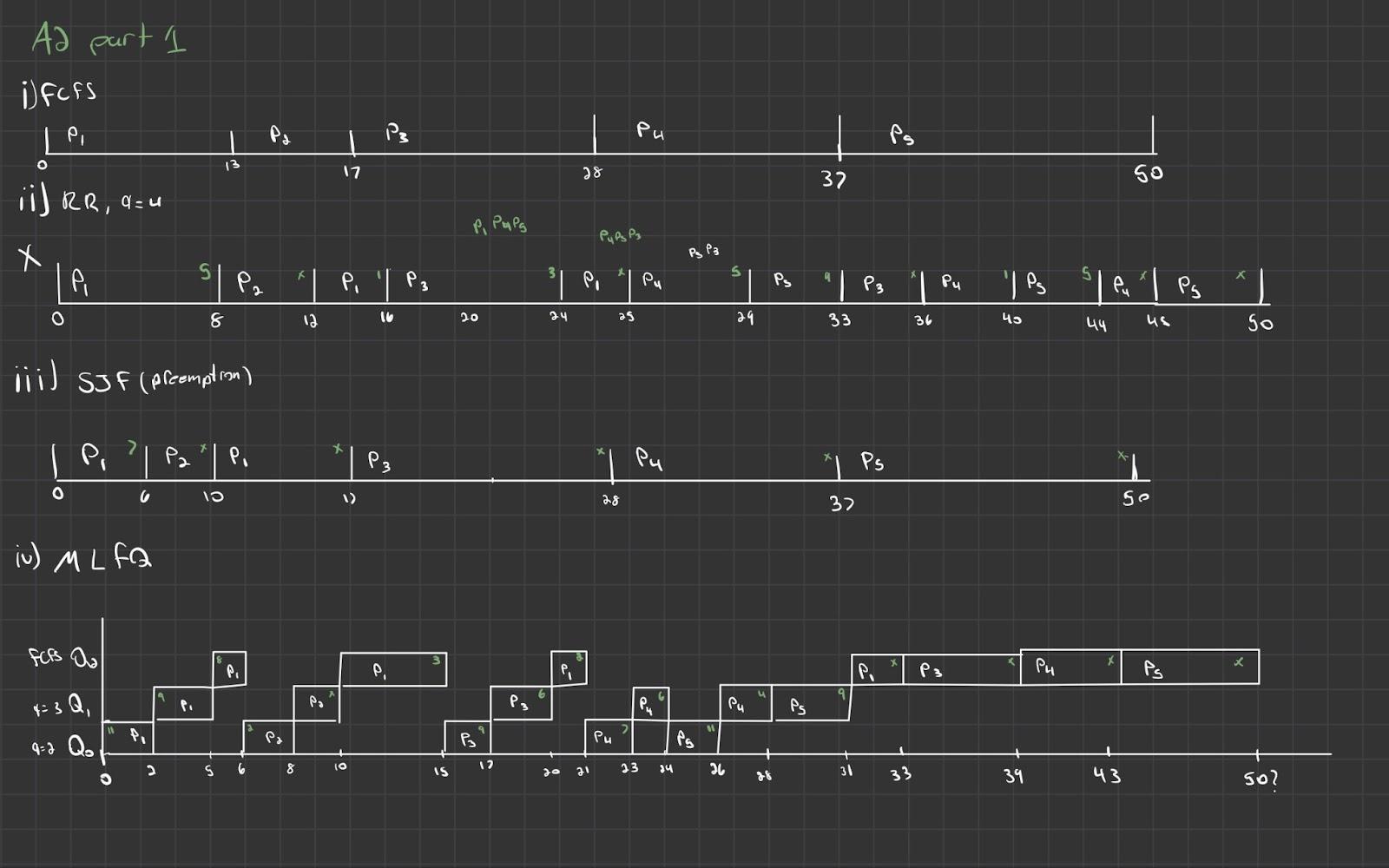
101259149

Assignment 2

SYSC-4001 section A

Gabriel Wainer

Nov 1st 2024



**C)**

### **First fit algorithm**

First Fit allocates the first hole that is large enough to hold the job.

#### Job 1 (122K):

* The first hole is 102K, not big enough
* The second hole is 205K, which is big enough.
* Allocate 122K to the 205K hole
  + 205K - 122K = 83K.

Updated list:  
102K, 83K, 43K, 180K, 70K, 125K, 91K, 150K

#### Job 2 (105K):

* The first hole is 102K, not big enough
* The second hole is 83K, not big enough
* The third hole is 43K, not big enough.
* The fourth hole is 180K, which is big enough.
* Allocate 105K to the 180K hole
  + 180K - 105K = 75K.

Updated list:  
102K, 83K, 43K, 75K, 70K, 125K, 91K, 150K

#### Job 3 (203K):

* The first hole is 102K, not big enough
* The second hole is 83K, not big enough.
* The third hole is 43K, not big enough
* The fourth hole is 75K, not big enough
* The fifth hole is 70K, not big enough
* The sixth hole is 125K, not big enough
* The seventh hole is 91K, not big enough
* The eighth hole is 150K, not big enough
  + No allocation possible for Job 3.

The list remains the same as the job was too big.

#### Job 4 (90K):

* The first hole is 102K, which is big enough.
* Allocate 90K to the 102K hole
  + 102K - 90K = 12K.

Updated list:  
12K, 83K, 43K, 75K, 70K, 125K, 91K, 150K

1. **Best Fit Algorithm**

Best Fit allocates the smallest hole that is large enough to accommodate the job. This minimizes wasted space.

#### Job 1 (122K):

* Available holes larger than 122K: 205K, 180K, 125K, 150K.
* The smallest hole that fits is 125K.
* Allocate 122K to the 125K hole
  + 125K - 122K = 3K.

Updated list:  
102K, 205K, 43K, 180K, 70K, **3K**, 91K, 150K

#### Job 2 (105K):

* Available holes larger than 105K: 205K, 180K, 150K.
* The smallest hole that fits is 150K.
* Allocate 105K to the 150K hole
  + 150K - 105K = 45K.

Updated list:  
102K, 205K, 43K, 180K, 70K, 3K, 91K, 45K

#### Job 3 (203K):

* Available holes larger than 203K: 205K, 180K.
* The smallest hole that fits is 205K.
* Allocate 203K to the 205K hole
  + 205K - 203K = 2K.

Updated list:  
102K, 2K, 43K, 180K, 70K, 3K, 91K, 45K

#### Job 4 (90K):

* Available holes larger than 90K: 180K, 91K.
* The smallest hole that fits is 91K.
* Allocate 90K to the 91K hole
  + 91K - 90K = 1K.

Updated list:  
102K, 2K, 43K, 180K, 70K, 3K, 1K, 45K

### **Worst Fit Algorithm**

Worst Fit allocates the largest available hole to the job to leave the maximum remaining space.

#### Job 1 (122K):

* The largest hole is 205K.
* Allocate 122K to the 205K hole
  + 205K - 122K = 83K.

Updated free list:  
102K, 83K, 43K, 180K, 70K, 125K, 91K, 150K

#### Job 2 (105K):

* The largest hole is 180K.
* Allocate 105K to the 180K hole
  + 180K - 105K = 75K.

Updated list:  
102K, 83K, 43K, 75K, 70K, 125K, 91K, 150K

#### Job 3 (203K):

* The largest hole is 150K.
* No allocation possible for Job 3.

#### Job 4 (90K):

* The largest hole is 150K.
* Allocate 90K to the 150K hole
  + 150K - 90K = 60K.

Updated list:  
102K, 83K, 43K, 75K, 70K, 125K, 91K, 60K

**Summary**

1. **First fit**
   1. This algorithm is the quickest and also the simplest. But there is the problem of memory fragmentation. Also, because the large hole was used for another job, job 3 was not able to be allocated.
2. **Best fit**
   1. It leads to jobs using space efficiently and leaving the smallest holes behind but that comes with the downside of leaving many small and more unusable holes (1k, 2k, 3k)
3. **Worst fit**
   1. This algorithm will try to avoid creating fragments in the holes. It will lead to leaving bigger holes to accommodate more jobs. But again like the first fit algorithm, it was not able to allocated memory for job 3

**Best algorithm**

* The best algorithm out of the three is the best fit algorithm. This method was generally good at conserving space and also reducing fragmentation in the holes. This is especially efficient if the jobs are varied in size
* The first fit algorithm is also performed very well but if fragmentation becomes an issue, that performance will significantly degrade.