CSCI 109: Introduction to Computer Science Homework 3
Yang Li

Problem 1: Programming
1. **(a).** 9
   (b). 30
   **(c).** a = $\log_b N$

Problem 2: Programming
2. **(a).** func(0): 1 times and print out 0
   func(1): 2 times and print out 1 and 0
   (b). The program does not run in polynomial time because it contains a factorial component, which would grow intractably fast.

Problem 3: Operating Systems
3. (a). First-come, first served (FCFS) processing:

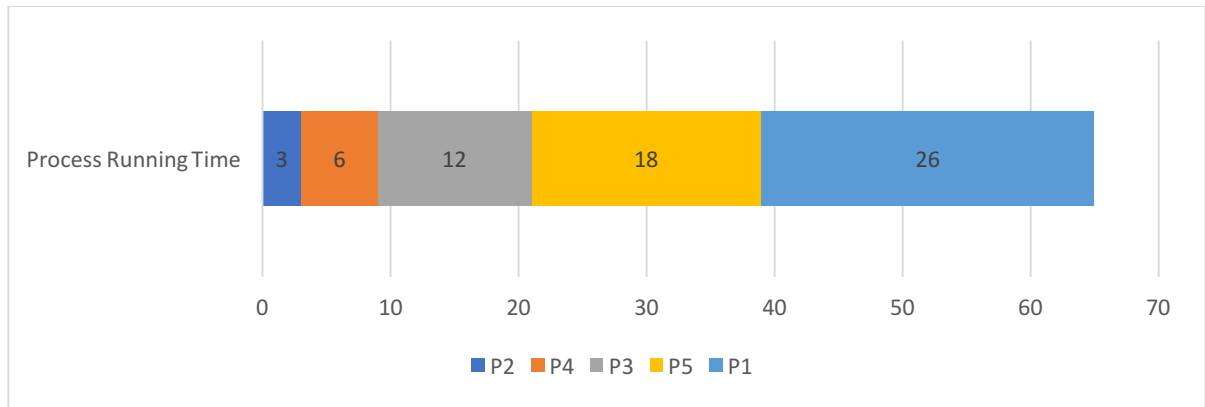| Order	Process	Process running time	Total running time
1	P1	26	26
2	P2	3	29
3	P3	12	41
4	P4	6	47
5	P5	18	65
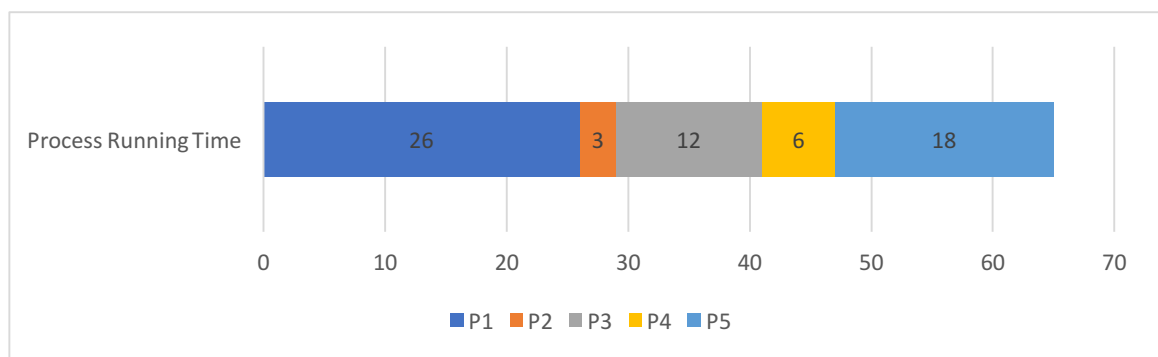
(b). P1 waits 0 unit of time.
   P2 waits 26 units of time.
   P3 waits 29 units of time.
   P4 waits 41 units of time.
   P5 waits 47 units of time.
   Average waiting time = (0+26+29+41+47)/5 = 28.6 units of time

(c). P2, P4, P3, P5, P1.

(d). Minimization of FCFS processing:

Order	Process	Process running time	Total running time
1	P2	3	3

2	P4	6	9
3	P3	12	21
4	P5	18	39
5	P1	26	65

Process Running Time: 3 | 6 | 12 | 18 | 26

P2 ■ P4 ■ P3 ■ P5 ■ P1

(e).
P1 waits 0 unit of time.
P2 waits 3 units of time.
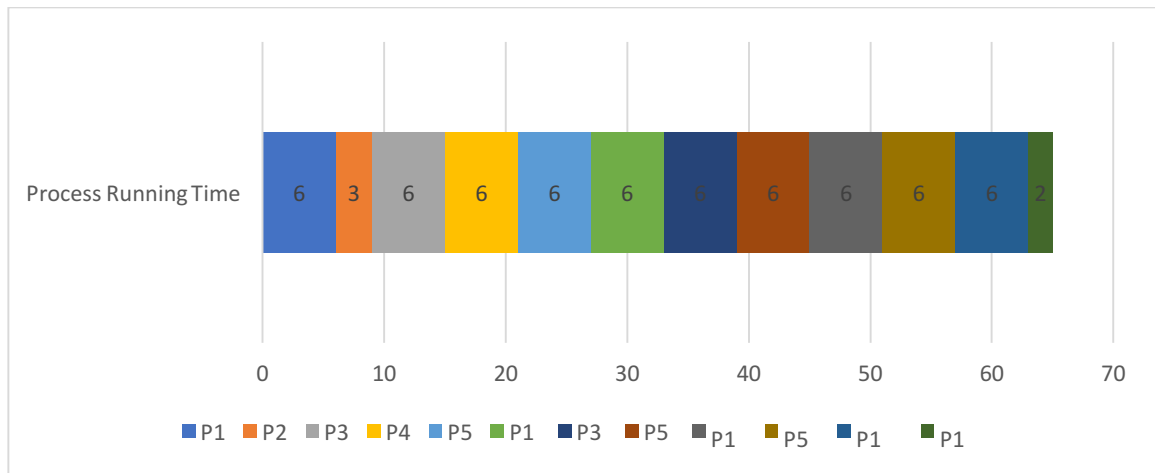P3 waits 9 units of time.
P4 waits 21 units of time.
P5 waits 39 units of time.
Average waiting time = (0+3+9+21+39)/5 = 14.4

(f). round robin scheduling:

Order	Process	Process running time	Total running time
1	P1	6	6
2	P2	3	9
3	P3	6	15
4	P4	6	21
5	P5	6	27
6	P1	6	33
7	P3	6	39
8	P5	6	45
9	P1	6	51
10	P5	6	57
11	P1	6	63
12	P1	2	65

P1 waits (65-26) = 39 units of time.
P2 waits (9-3) = 6 units of time
P3 waits (39-12) = 27 units of time
P4 waits (21-6) = 15 units of time
P5 waits (57-18) = 39 units of time
Average waiting time = (39+6+27+15+39)/5= 25.2 units of time

(g). $(n-1)*q$

(h). Advantages:
- Short waiting time.
- Fairness is promoted as every process gets equal units of time.
  Disadvantages:
- Program has to be frequently interrupted, which lowers the efficiency of CPU.

(i). When a user program needs access to protected resources it makes a system call (e.g., managing files, accessing a printer). Therefore, OS would have very high privilege in running programs so that crash in one program doesn't necessarily crash OS and other programs.