

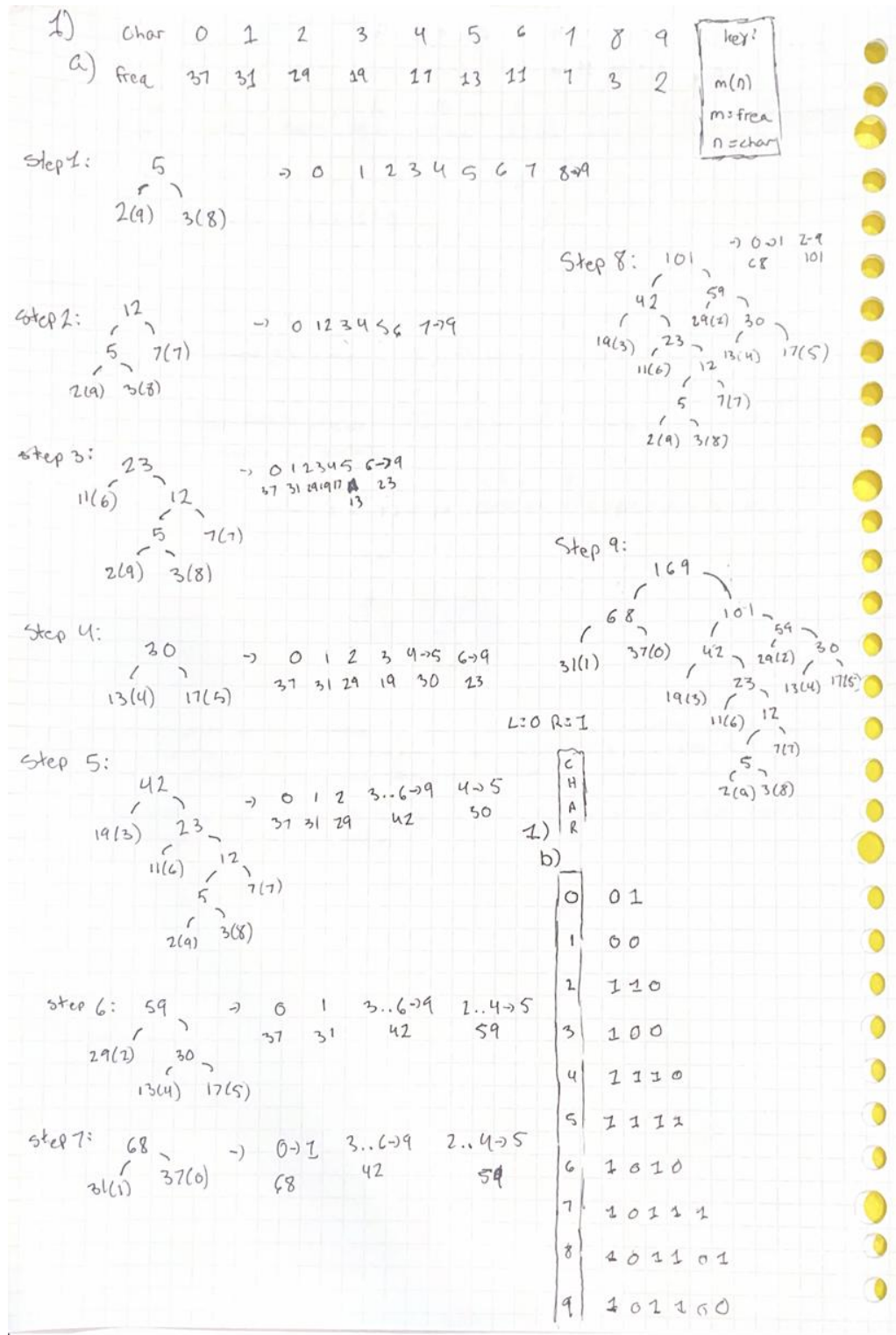
CS444: Operating Systems

Homework 1

Due: 2/13/2025

Name: Sawyer Bowerman

1. Huffman Coding: Figure 1: HW1 Question 1a & 1b



2. Chapter 1 metric prefixes and ASCII codes for project1:

- a) There are 1,000,000,000 nanoseconds in 1 second. $\rightarrow 1 \text{ nanosecond} = 1 * 10^{-9}s$
 b) 1 gigabit = 1,000,000,000 bits per second $\rightarrow 1 \text{ gb} = 1 * 10^9 \text{ bits}$
 c) 1 kilobyte = 1024 bytes, 8bits wide $\frac{1024}{8} = 128 \text{ bytes tall}$

'A':

- Dec: 65
- Bin: 01000001
- Hex: 41

'9':

- Dec: 57
- Bin: 00111001
- Hex: 39

'?':

- Dec: 63
- Bin: 00111111
- Hex: 3F

3.

Figure 3: hw1 Question 3a-3b Answers

3)

a)

i) Shortest-Job-First : $T = \begin{matrix} 0 & 3 & 6 & 8 & 9 & 12 \end{matrix}$
 $P2 \rightarrow P1 \rightarrow P4 \rightarrow P6 \rightarrow P5 \rightarrow P3$

ii) Non-Premptive priority : $T = \begin{matrix} 0 & 3 & 6 & 10 & 13 & 14 \end{matrix}$
 $P2 \rightarrow P1 \rightarrow P3 \rightarrow P5 \rightarrow P6 \rightarrow P4$
 (Lower ~~is~~ = higher priority)

iii) RR $T = \begin{matrix} 0 & 2 & 4 & 5 & 6 & 8 & 10 & 12 & 14 & 16 \end{matrix}$
 $P1 \rightarrow P2 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P3 \rightarrow P5 \rightarrow P6 \rightarrow P5$
 Not certain this is right...
 1 left 1 left

b) The priority inversion problem is when a thread is occupying the space when one of higher priority is waiting to be processed. This could cause for inefficiencies as a process stuck in this state could possibly never move from this current state.
 According to the slides, the solution is that the high priority thread should be put to sleep instead of "busy waiting."