**Ch 1. Ex. 5**

Without having to get into code and actually writing out the programming logic, I can only really explain the answer to this problem in the following way:

Basically, if a ship is coming into port the next day and that port is empty, then the ship will remain there. If a ship is out and its’ intended port is not empty, then that ship will move to port + 1 until it finds an empty port to remain.

**Ch 2. Ex. 2**

The answers for input sizes corresponding to each run time in question are listed on the accompanying excel spreadsheet.

NOTE: For run times: n log n, and 22^n

For n log n, the calculated input size exceeds the 1 hour time limit for operations and run time. I’m not sure how to accurately calculate the input size for this problem, but I was close? In solving for input size, the results overshoot the one hour time limit.

For 22^n I used a similar approach as the previous question. Used logs to solve for input size, but this did not yield legitimate results. Not sure where I’m messing up there.

**Ch 2. Ex. 3**

f2 < f3 < f6 < f1 < f4 < f5

**Ch 2. Ex. 6**

1. Outer loop takes constant time over *n* O(n), inner loop takes constant time over *n* elements O(n). So, upper bound would be O(f(n2)). Also, more simply put, from earlier in the chapter on the section in quadratic time, the authors mention nested loops and the effect it has on run time.
2. The outer for loop has to execute at least n times, and so will the inner loop to populate the B array. This results in lower bound of Ω(f(n2)).
3. I tried a few different pseudo code algorithms and I can’t figure out a way to reach the end goal without using nested for loops. This leads me to believe that I a.) got the first two parts of this problem wrong, or b.) am thoroughly underprepared for this class. This is disheartening as I aced a semester of 400 level CIT courses and this 200 level course is proving pretty difficult for me to grasp.