**Brandon Bardwell**

**Discrete Structures**

**(Date)**

Section 3.1, Number 7  
Section 3.2, Number 7  
Section 3.3, Number 15

**What We Learned or Noticed:**

**Problem 1:**

**7.** Describe an algorithm that takes as input a list of *n* integers

and finds the location of the last even integer in the

list or returns 0 if there are no even integers in the list.

**Sentence:**

**References:**

Pg. 192. Example 1

**Work:**

1. Set the temporary address of the last even integer in the sequence as 0.
2. Compare the next integer in the sequence. If it is even, change the address to match the current integer’s address. (Starting with 1 for the first number)
3. Repeat the previous step if there are more integers in the sequence.
4. Stop when there are no integers left in the sequence. The last even integer in the sequence is at the address given. If the address is still 0, there were no even integers in the sequence.

**Discussion:**

It would probably be better to store -1 as the address for no evens, because -1 can’t be stored in an array unlike 0 can.

**Problem 2:**

**7.** Find the least integer *n* such that *f (x)* is *O(xn)* for each

of these functions.

**a)** *f (x)* = 2*x*3 + *x*2 log *x*

**b)** *f (x)* = 3*x*3 + *(*log *x)*4

**c)** *f (x)* = *(x*4 + *x*2 + 1*)/(x*3 + 1*)*

**d)** *f (x)* = *(x*4 + 5 log *x)/(x*4 + 1*)*

**Sentence:**

**References**

**Work:**

**Discussion:**

**Problem:**

**Sentence:**

**References**

**Work:**

**Discussion:**