**MIDTERM 3 REVIEW QUESTIONS**

**Programming questions: all on sorting template arrays, like the examples from class.**

1. **Reverse the order of any of the 6 sorting algorithms.**
2. **Make mergesort (or quicksort) use pointer arithmetic to define its subarrays.**
3. **Write the partition function for quicksort assuming no duplicates.**
4. **Extend the partition function for quicksort to handle duplicate pivots.**
5. **Write me the loop that builds a heap from an array of n numbers in heapsort.**
6. **Write me the loop that removes items from the heap to the end of the array and re-heapifies.**
7. **Make quicksort use insertion sort on subarrays of size 12 or less.**

**Written questions:**

1. **Show me how the type of a template class gets defined in the main() program.**
2. **How would you define an array of big\_numbers using a template array class like the ones from the sorting codes?**
3. **How could you count the frequency of a number in a binary search tree?**
4. **How could you tell if 2 binary search trees contained exactly the same numbers?**
5. **Could we make a template class for binary search trees? Why or why not?**
6. **What’s the expected run time for quicksort? Why? What’s the worst run time it can get, and when does this happen?**
7. **What’s the expected run time for mergesort? Why? What does mergesort do that’s inefficient compared to the other sorting algorithms?**
8. **What’s the best case run time for insertion sort? What produces this run time? What’s the expected run time for insertion sort? What about bubble sort and selection sort?**
9. **Convert 23415 in base 7 to base 5, using the digit-wise algorithm from class.**
10. **What is the problem with this code?**

**int& no\_no\_nanette() {**

**int answer = 9;**

**return answer;**

**}**

**Why is that not a problem with this code?**

**big-number& operator =(const big\_number& m) {**

**…**

**return \*this;**

**}**

1. **Tell me 2 big differences between a copy constructor and an assignment operator (operator =). Justify your answers.**
2. **Given the code for operator >, what could you do to get operator < for very little work? How would you get operator == from the > and < operators?**
3. **For the pattern code, how many pattern calls result from calling:**

**pattern(outs, 4, 0);**

**pattern(outs, 16, 0);**

**pattern(outs, 1024, 0);**

**What formula describes this relationship between the starting n and the number of calls pattern makes?**

1. **What formula describes the relationship between the starting n and the number of stars pattern prints?**
2. **Given the array below: 6 3 9 8 8 3 1 7 3 9 1**

**Show me the array after one pass (one inner loop) of selection sort has run.**

**Show me the array after one pass (one inner loop) of insertion sort has run.**

**Show me the array after one pass (one inner loop) of bubble sort has run.**

**Show me the array after one partition step of quicksort has run. Which subarrays will the code sort next?**

**Show me how mergesort will process this array to sort it.**

1. **Draw me the heap you get from adding the numbers 9, 4, 5, 3, 2, 7, 8, 7 to an empty heap.**
2. **Draw me the heap you get from adding the numbers 3, 9, 7, 2, 7, 8, 5, 4 to an empty heap.**
3. **Draw me the heap you get after removing the 9 from the heap in the previous question.**
4. **Show me the code that checks for self assignment in operator =, and tell me what it’s checking.**

**Where else in big\_number might we need to check for this?**

1. **What does partition do to make the array ‘less unsorted’ than before?**
2. **Given an array representation of a heap, tell me a formula to get the parent of a heap item in the array. Tell me a formula to get an array heap item’s left child, and its right child.**
3. **Given the answers to the question above, write me a loop that re-heapifies a heap after one new element has been added.**
4. **Given the answers to the question above, write me a loop that re-heapifies a heap after the root element has been removed.**
5. **What is particularly good to eat around Boulder? Delight the tastebuds of Dr. White.**