Final Exam Study Questions

(Study everything, some maybe most of these questions will be on the exam.)

1. **Study everything on exam #2** except the linked list and queue questions. Similar types of questions will be on the final. Insure that you know these – especially if you got them wrong on the exam.
2. Stacks and Queues question: using an example STL Queue and/or Stack, given a set of operations, indicate the final elements of the queue or stack
3. The final exam may also have questions on the following:

-write the C++ code for a binary search (you have an ordered array)

-Binary Search tree (slideset 10b), how to create, add and delete items , e.g.

given values, create a binary search tree

add a value to a given binary search tree

delete a value from a given binary search tree

-define and give example of a complete binary tree

-define and give example of a heap

-Know how to insert, delete items from a heap

-Given a heap as a complete binary tree, show the implementation in an array

3. **questions on hash tables, example:**

-show a hash table of size 9, and insert the keys 5, 29, 20, 0, 18 into the table using the hash function, key%9

-a *chained* hash table has size 500, how many keys can be inserted?

a. 500

b. 501

c. 1000

d. there is no maximum

1. **Running time**
2. For an unordered array of n elements, which of the following is worst-case running time for the search?
3. O(n)
4. O(log n)
5. O(1)
6. None of the above
7. For an ordered array of n elements, which of the following is the worst-case running time for the search?
8. O(n)
9. O(log n)
10. O(1)
11. None of the above
12. For a hash table, which of the following is the following is the worst-case running time of the search?
13. O(n)
14. O(log n)
15. O(1)
16. None of the above

**Linked list questions**

Two of the following will be on the exam

-Algorithms only

-Note: you may assume a tail pointer

You can consult with others (who know this), look them up online, etc. But you must know these, trying to memorize them will not work. Remember the boundary conditions discussed in class.

1. Search a linked list for a specific value and if found, remove it from the list. If not found, return 0.
2. Count the number of times a specific value occurs in a linked list and return the number. If not found, return 0.
3. Given a non-empty linked list of integers that includes a node with value 10, move that node to the end of the list.
4. Given a non-empty linked list, exchange the last node with the first node
5. Reverse the elements in a linked list

Notes:

-these are singly-linked lists

-if the list is empty, head=null and tail=null

Final linked list question, using the following code which will be on the exam.

Linked list definition (singly linked only)

struct node{

{

int data;

node \*next;

}

class linked\_list

{

private: \*head, \*tail;

public:

linked\_list()

{

head = NULL;

tail = NULL;

}

1. Add a function that adds a new node to the linked list as described above.
2. Add a function to print out the values of a linked list as described above.

**The answers to these questions are online,** look them up, read and understand them, be prepared to provide them on the exam.