Final Exam details

The final exam usually has about 9 questions and is worth a total of 100 marks. There are no multiple choice questions, and no calculator or notes are allowed to be brought in to the exam.

The following are a small selection of questions from past exam papers. **It is not** guaranteed that similar questions will appear in future exams, although the style of question (short coding questions) will be similar. Questions that are almost certain to appear on a final exam include counting the number of assignments/comparisons in a given snippet of code, and knowing/understanding the asymptotic complexity of various operations on the various data structures we studied. The best revision sources are the COSC230 lectures, practicals, and assignments.

Question 1 [*12 marks total*]

The following `delete_from_tail()` member function from the class `SLList` implementing a singly linked list of integers contains some errors:

```
int SLList::delete_from_tail()
{
    int a = tail->info;
    if (head == tail) {
        delete head;
        head = tail = 0;
    }
    else {
        SLLNode *p;
        for (p = head; p->next != tail; p = p->next);
        delete head;
        head = p;
        tail->next = 0;
    }
    return a;
}
```

(a) Draw a sequence of three or more diagrams to show how this operation should work on a singly linked list with more than one node.

(b) List and fix each error in the `C++` code for member `SLList::delete_from_tail()`.

Question 2 [*8 marks total*]

Write a recursive function that produces the same output as the following iterative function:

```
void cubes(int n)
{
    for (int i = 1; i <= n; ++i) {
        cout << i*i*i << " ";
    }
}
```

Question 3 [*14 marks total*]

Draw a diagram of the binary search tree that results from inserting nodes with the following values: 15, 20, 4, 16, 25, 1, 8, 7. Next, write down the order of nodes visited in the following traversals:

(a) Top-down, left-to-right, breadth-first traversal.

(b) Inorder, depth-first traversal.

(c) Preorder, depth-first traversal.

(d) Postorder, depth-first traversal.

Question 4 [*12 marks total*]

Use the Quicksort algorithm pseudocode you covered in Practical 7 to sort the following list into ascending order: 18, 20, 11, 7, 17, 15, 14, 21, 23, 16, 9. The bound (pivot) can be chosen as the first element of a list and is not included in the derived sublists, and partitions can be made by using $\leq$ and $>$ in a consistent manner. Be clear about writing your bound and lists at each step.