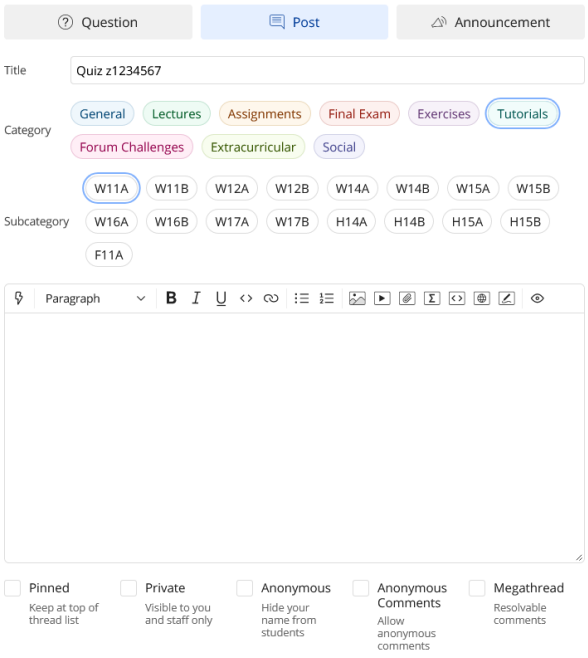
In this quiz, we review some content from COMP2521/9024, and establish the standards for describing and reasoning about algorithms.

You can discuss these problems with anyone, and you can use any resources to solve the problems, as long as you write in your own words. You have to submit your first attempt to complete the quiz by Friday of Week 4; you have to submit your final version which incorporates your changes following the feedback from your tutor by Friday of Week 7.

You'll submit your attempts by making a post on the Ed forum:

- click 'New Thread' in the top left corner
- select 'Post'
- title your post 'Quiz z1234567', using your zID
- select the category 'Tutorials'
- select the subcategory corresponding to your tutorial code
- enter the text of your responses
- select 'Private'
- click 'Post'



You can insert images and equations into the text box. We ask that the text of your response be typed into the text box rather than attached in an uploaded file.

Please only make **one** post to answer all questions. **Do not** make a separate post for each question.

After posting, you can always edit the post or add comments if you'd like to amend it before receiving feedback.

Once you make a first attempt, your tutor will make a comment below it, using a 'traffic light' system.

- Green means you've completed the task.
- Yellow means minor revisions are required.
- Red means major revisions are required.

If the task isn't complete, your tutor will leave additional comments specifying what you have to change. You should then edit your original post or add further comments to address this feedback, and your tutor will re-assess the work. **Do not** make a new post for a second attempt.

You must eventually receive 'green' feedback to complete the task. This will earn all five course marks, regardless of the number of attempts.

We suggest that you make a first attempt before your second tutorial, but this isn't a hard deadline. If you don't know how to do a problem, write that in your post and your tutor will give you some help in their first round of feedback.

This must be completed by the end of week 7, but we anticipate that most students will achieve this in the first two or three weeks of term.

Aside from this assessment, you can should also post to the Ed forum about tutorial discussion points and problems, as well as practice problems. Those posts should be public rather than private, so that other students can see and interact with them.

Question 1

Suppose you have a binary search tree in which each node stores a distinct integer. If a node contains the value x, we will say that the node with least value greater than x is its *successor*.

Explain why if a node containing x has a right child, then the successor of the node containing x does not have a left child.

Hint: Show that the successor of x is the minimum element y of the right subtree of x. Consider all other candidates z for the successor of x which are outside of the subtree rooted at x; argue that x and y can be both $> z$ or both $< z$, but that $x < z < y$ is impossible.

Question 2

2.1 Define what a complete binary tree is.

2.2 Define what a heap is.

Suppose you have a max heap in which each node stores a distinct integer. Consider the following algorithm which claims to implement the 'pop' operation, i.e. remove the node of largest value from the heap.

> Maintain a pointer to the node of largest value, initially at the top of the heap, and recurse downwards. At each stage:
>
> - if the node has two children, swap it with the child of larger value;
> - if the node has only one child, swap it with that child;
> - if the node has no children, delete it.

2.3 Demonstrate the execution of the algorithm on a nontrivial example in which it correctly removes the node of largest value from the heap.

2.4 Does this algorithm correctly implement the 'pop' operation? Justify your answer.

> Hint: 'pop' operation must preserve the complete binary tree shape of the heap.

2.5 Describe how to correctly implement the 'pop' operation.

Question 3

A conveyor belt contains n packages that need to be delivered to AlgoTown within K days. Package i has an associated integer weight w_i, where no package has weight exceeding a maximum M. Each day, the packages are loaded onto a truck in the order of their position on the belt. We may not load more than the capacity of the truck, which is also an integer.

3.1 To get a better understanding of the description above, consider the following example. There are $n = 10$ items with the following weights: $w = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ and $K = 5$.

- Determine if it is possible to load all packages within 5 days if the truck's capacity is 12.
- Determine if it is possible to load all packages within 5 days if the truck's capacity is 17.

3.2 Explain why it is always possible to load all packages within K days if the capacity of the truck is equal to the sum of the weights of the packages, regardless of the value of K.

3.3 Given the capacity C of the truck, the weights of the packages and the number of days K, design an $O(n)$ algorithm that checks whether it is possible to deliver all packages within K days.

3.4 Suppose now that you get to choose the capacity of the truck. To save money, you would like to choose the smallest possible truck that allows you to deliver all packages within K days. Explain why binary search is a feasible strategy to solve the problem.

> Is there perhaps a monotonic behaviour hidden in the problem?

3.5 Hence, design an $O(n \log(nM))$ algorithm that solves the problem in part 3.4.

> This problem is discussed in Tutorial 1. Write up the solution outline presented in the tutorial in your own words.

> In parts 3.3 and 3.5, you must also justify the correctness and time complexity of your algorithm, as with every other algorithm design problem in this course.