

Workshop 4 – Week 5 – Worksheet 5

Question 5.1 A stack is an abstract data structure that behaves much like a stack of books on a desk: The newest additions go on top, and the newest additions are also the first to come off the stack. Show:

1. How you could implement a stack using an array as the underlying data structure. Think about how you would deal with the situation where the stack is full.

Items could be inserted at the next free space in the array, which could be a pointer stored in the data structure. When the stack fills, it could be reallocated to add additional space, as this implies, the current stack size limit would need to be stored in the stack structure as well.

2. How you would implement the stack using a linked list as the underlying data structure.

Items could be inserted at the start of the linked list. Alternatively, a pointer to the end of the linked list could be given and a doubly-linked list could be used. The first option is quite probably the better one.

Question 5.2 A queue is an abstract data structure that behaves much like a queue in a grocery store: The newest additions to the queue go at the end (tail), and items are taken out of the queue and processed from the front (head). Show:

1. How you could implement a queue using an array as the underlying data structure. Think about how you would deal with the situation where the queue is full. Discuss whether a circular array can be used.

An array could be used and each time an item was removed (or when there was room but the end of the array had been reached) and then all items could be moved forwards until hitting the front of the array. When this type of queue fills, more space could be allocated. With a circular array, a pointer (in the form of two array indices) could be kept to the start and end of the array, when an item is inserted it is inserted at the end of array pointer, and then that pointer is advanced, if the array fills, all items can be moved to the front of the array (that is, moved so that the start of the queue pointer points to the start of the array) and then the array can be extended to allow more items to be inserted. Other options exist, such as making a new queue and taking all the items from the old queue and putting them into the new queue (taking the first item off the first queue and then adding it in the second queue).

2. How you would implement a queue using a linked list as the underlying data structure. Discuss the pros and cons of the linked list and array implementations.

The linked list means that items don't need to be moved forward at all, but involves many more memory allocations.