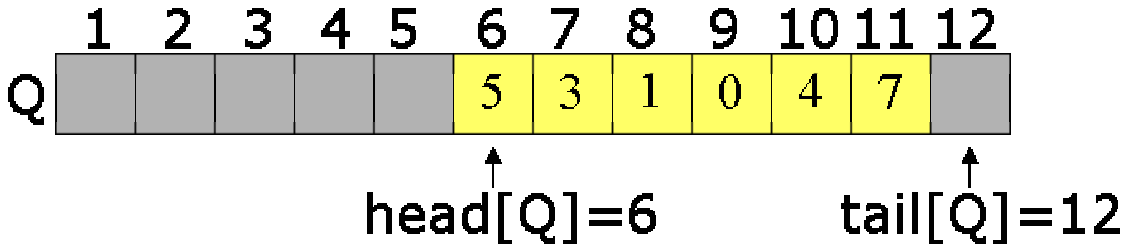
Lecture C4: Stacks and Queues

Response to 'Muddiest Part of the Lecture Cards'

(19 respondents)

1) *What is "implementation specific exception", you said it happens when trying to push a new element onto a full stack*?
It is implementation defined. In our case with Ada, if you try to add a value to an already full array, an exception will be raised (runtime error / constraint error). On the other hand, if you wrote a better version of your program you might have built in error checks to see that out of bound operations on the array never takes place, maybe instead a "friendly message" will be printed to the Monitor telling the user that the stack/array is full.

2) *When a queue is implemented as an array, what happens when tail[Q] = head[Q]? What if tail[Q]=2, head[Q]=3, you add one, and then want to add another element without taking any out*?
If you use the same definition of the queue presented in class, then head[Q] is always less than tail[Q].



Head[Q] points at the next item that will be removed/served from the queue. Tail[Q] points at the first empty place in the queue is, that is, where next item being enqueued will be placed.

If the queue is circular, then when tail[Q] = head[Q], then the queue is either empty or full.

If tail[Q] = 2 and head[Q]=3, means that the queue has one empty spot, next enqueued element will be written to position 2 in the queue, after that tail will be increased by 1 and have the same value as head, that is the queue is full. If you now want to add one more element, then it depends on how the 'enqueue-operation' is implemented. One alternative is not allow the enqueueing of a new element, but print an error message, or maybe just overwrite the last position in the queue with this new element … or some other solution that seems feasible for your problem.

3) *Performance: each op used is O(N), what does that mean*?
"Big-O" is used within 'complexity theory' to denote an asymptotic upper bound for the

magnitude of a function in terms of another usually simpler function. Much more information about "Big-O" will be covered in lectures 9, 10 and 11.

4) *What is the corresponding "push" and "pop" commands for a queue*?
Enqueue adds an element to the queue and *Dequeue* deletes an element from the queue.

5) *Head[Q] and Tail[Q] are pointers, right*?
Conceptually they are pointers, but according to the implementation, they are indices to the array. The queue has an attribute head[Q] that is the position of the first element in the queue, and tail[Q] points at the position where next element will be inserted.

6) *When the number 2 was popped from the stack, you said it was still in the array, but not the stack? Can you still reference the 2? For Example, If you did push(S,x), what would happen to the 2*?

The element will still be in the 'memory location' until a new write/push operation overwrites the memory locations with a new element.

No, you cannot reference the 2 after it has been popped. There is no guarantee that no other process in the computer has not used the stack and thus overwritten the value 2 that used to be on top of stack, before you try to reference that memory position.

If on the other hand, if you did a push, the element 2 is overwritten with x.

7) *If the queue wraps around and fills up, and user wants to enter a element at 5, is the element overwritten or does an error message result? Or is this completely up to the programmer of the package*?
That is up to the programmer.

9) *How did they come up with the idea of postfix*?
Postfix notation (also known as RPN / Reverse Polish Notation) has the advantage of being very easy for a computer to analyze. Compare the algorithms in lecture notes for evaluating postfix vs. infix expressions. RPN was invented in 1920 by a Polish mathematician Jan Lukasiewicz. Take a look at http://www.hp.com/calculators/articles/rpn.html for more information on why one should concider using RPN.

10) *What is an HP-calcualtor*?
A calculator made by Hewlett-Packard. HP produces a number of calculators that uses RPN (Reverse Polish Notation http://www.hp.com/calculators/articles/rpn.html) Why not take a look at the following web page: http://www.hpmuseum.org/ or this one: http://www.hp.com/calculators/

11) *In postfix, how does it know which order to perform - /* ?
All operators have a known precedence and also known how many operands the operator needs. To evaluate a postfix expression, use the algorithm shown in class. Each time an

operator is read, pop the correct number of operands from the stack, and push the intermediate result back on top of the stack.

12) *No mud and happy comments/pictures of 'spring break'* (9 students)