Tyler Hackel

Professor Turnage

CIS 2723

Chapter 7 Homework

Stacks

* Stacks can be viewed as a pushdown method or a last-in-first-out (LIFO) data structure.
* This type of structure operates on the same principle as a Pez Dispenser – used for storing candy. To fill a Pez, you have to insert the candy from the top (push). As the candy is being pushed down, the last piece of candy inserted will be the first one removed (popped), hence the LIFO structure.
* This structure of stacks is integrated with the doubly linked list method. While adding new data, we simply rename the head of the linked list to be at the top. When removing data we access the head, or last data added, to retrieve the following data in the stack.
* In relation to the pop and push method is the peek method. This allows the user to retrieve the data information without having to remove a link or change the top value.

Queues

* In contrast, queues use a systematic structure of the first-in-first-out (FIFO) principle.
* This type of structure operates on the same principle as fast-food restaurants. When food is ordered it is placed at the end of the list (enqueue). As the orders are made complete, the remaining orders become closer to the front of the list and popped from the list (dequeue), hence the FIFO structure.
* The head and tail value change when data is added to a queue-structured linked list. If data is placed in enqueue, the new data becomes the value of tail. Oppositely, if data is placed in dequeue, data next in queue becomes the value of head.

Priority Queues

* Priority Queues are similar to queues but offer a managing feature that implements items of priority.
* This structure maintains the same principle of FIFO. Although, data is sorted by highest priority when added as the tail. This feature allows new data to be executed based on its importance, rather than the order of arrival.

The Heap

* The heap is a structure that allows priority queues to be maintained more efficiently. For example, the Heap uses a top-down effect that orders data in a branching method. This method of branching, similar to a tree, allows faster access to data access. This increased efficiency is possible because the heap doesn’t need to access all links, only the links of higher priority.

|  |  |
| --- | --- |
| STACKS | QUEUES |
| sTACKING DISHES ON TOP OF EACH OTHER | A TICKET LINE |
| tHE JAVA VIRTUAL MACHINE | AN ESCALATOR |
| THE UNDO BUTTON IN MICROSOFT WORD | A CAR WASH |
| CONTINUING A SAVED VIDEO GAME | PRINTER QUEUE |
| TENNIS BALL CONTAINER | KEYBOARD STROKES |