Incorporating new experiments for Data Structures Lab: a prototype for Queue Simulation

Author: Jayashree Prasad
Date: 20-04-2014

Currently the D.S lab doesn't have experiments pertaining to:

1. Stack operations
2. Queue operations
3. Singly linked List implementation
4. Doubly linked list implementation

(However there are experiments which show the application of these D.S's. Eg:- expression evaluation(makes use of stack) polynomial addition (makes use of linked list) etc)

IMO, not just Engg. students, but even people who are wanting to acquire a basic knowledge of Computer Science are likely to be interested in Data Structures Lab. In this regard adding more experiments to this lab will be helpful for end users. Probably it will be good to include the above mentioned experiments, which illustrate core D.S.

Queue using linked list:

- This is a new simple simulation which I've done using HTML5 (canvas) & JavaScript The simulation demonstrates the 2 operations of a queue: (1) insert (at head) (2) delete (at tail)
- isFull, isEmpty operations are taken care, through status bar displaying the appropriate status. (The number of elements in the queue.)
- Insert and Delete operations are animated using setTimeout API. (I think this Delete animation is nice :-)
- The head and tail pointers are shown in every frame along with the movement of the queue. (Every operation needs redrawing of the canvas frame completely)

My comments & suggested improvements for this implementation:

- Implementation is based on MVC. (although nothing much to model in this case)
- Code needs to made more modular still; need some more re-organisation. (when to have classes & when to have free functions is a design decision one needs to make.)
- Code needs alteration to make it responsive; adding appropriate CSS.
- Based on the screen size, and the start position of a node, the positions of all other nodes can be calculated using generalised equations; as of now have calculated it statically.
- IMO, the black-red color combination of the nodes look good.
- The head, tail pointers and the horizontal pointers look decent.

- However the input section and the status bar section can be made more attractive/ better with some more thought.
- A section which displays a trace of pseudo-code too can be added if required. (just like the trace for CP labs)
- If required, New experiments *viz Stack operations,Singly linked List implementation, & Doubly linked list implementation* can now be implemented easily based on the Queue code.

***