Comp 2071: Data Structures, Spring, 2016

Terrance Curley

Rixing Wu

Elvin Xu

Gregory Lee

Lab 2

Group 16

Due: 2/2/2016

Analysis:

1. “In general, how useful is the Stack data structure? Can you think of other uses aside from those discussed in the applications chapter?”

The Stack data structure is useful in a multitude of situations, because many applications can utilize the last in first out process. It is a concept that is common in many programs. As with all data structures the Stack seems to have its place, and can be the best options for certain programs. It is not useful in every application, however.

One potential use of a stack that I see in many programs is to store edits to a file that can be undone. This is a widely used concept throughout text editors, document writers, and image editing programs. When reversing edits to a file it makes the most sense to undo the most recent change first. Therefore, you could store a history of edits, like words typed or lines drawn, by pushing them onto a stack. Un-doing one of the edits could pop this change off of the stack, allowing you to undo changes made previously. This is just one of many possible implementations.

2. “If we didn’t have the Stack data structure can you think of another way to implement the LIFO (last in first out) concept?”

    The LIFO concept can be implemented in many different ways. One way is to use arrays. A Last in first out structure could be implemented by storing data in an array, increasing the index with each piece of data stored. When removing this data you could limit access to the largest index being occupied. This would effectively make it so that the last item stored is the first item retrieved. Thus, you have a LIFO data structure. This is how we created our ADT.

Another way to implement the LIFO concept could be a linked list of nodes. You can store data in these nodes and have each node point to the previous node. In this sort of structure, the most recently stored data would be the head of the linked list. If you wanted to retrieve data you could remove the head and then have the node that it points to become the head. This would be similar to a stack.

Summary:

1.

a. “How did you “divide up” the work so that each student still met the objectives for the assignment (i.e., learn, understand and apply the concepts)?”

    For this Assignment we split up the work so that the work would be split up among the group. Rixing worked on the application and Gregory did the graphical interface, Elvin worked on the Stack ADT, Terrance wrote the writeup, and the group as a whole organized the presentation.

    In order to ensure that each group member understood the material, we collaborated as a group multiple times and all helped each other with our respective parts of the work. In order to make the presentation clear we had to be sure that each of us knew the material, so we each talked about the work we had done in our group meeting.

b. “How did you coordinate code changes/testing?”

As mentioned, we talked in person as a group. We tested the program and gave advice to each other on how to improve our different parts. If we made changes to our code separately we could text and use github to share our code with each other.

c. “Other observations about working with a partner?”

One observation about group work for this assignment is that splitting up the work can make for a better experience overall. Not only is it less stress for everyone, but when we specialize on different aspects of the application we can help each other understand certain things about the way the application works. This helps us further our understanding of the stack data structure.

2. “Where did you have trouble with the assignment? How did you move forward? What topics still confuse you?”

Elvin had trouble allowing his array-based Stack to re-size itself if it reached capacity. This is important for the stack to ensure that it can be dynamically sized. The solution to this is to create a larger array and copy to contents of the full array over. This can be tricky but with help of the group and by using the textbook as a resource we were able to fix this aspect of our data structure.

3. “What did you learn from the assignment? (Please be specific.)”

Terrance: I had learned about and used the stack data structures in previous courses, but I don’t think I fully understood the concept until this assignment. Making a stack ADT from scratch really helps you understand how it works conceptually. In particular being able to see it in an array implementation made things more clear to me.

Elvin: The ADT implementation for stack allows me to better understand how methods like push(), pop(), peek(), perform and work under the hood.

4. “How could this assignment be improved in the future?”

  I think this assignment was helpful overall. It may be beneficial for the assignment to cover multiple implementations of the stack ADT, and how an array-based implementation compares to a linked list or a vector. This was covered well in class, however.