Worksheet 24 - Group 1

Worksheet Group 1 Members

Marc Clinedinst: clinedim@onid.oregonstate.edu
Kelby Faessler: faesslek@onid.oregonstate.edu
James Fitzwater: fitzwatj@onid.oregonstate.edu
Tom Gariepy: gariepyt@onid.oregonstate.edu
Sean Reilly: reillys@onid.oregonstate.edu
Joseph Struth: struthj@onid.oregonstate.edu

Collaborators

Marc, Kelby, James, Tom, Sean, Joseph

Worksheet 24: Linked List Iterator

In this assignment, we implement four different functions which control the behavior of an iterator used to iterate through the values stored in a linked list. More specifically, we implement the `linkedListIteratorInit`, `linkedListIteratorHasNext`, `linkedListIteratorNext`, and `linkedListIteratorRemove` functions. These functions are described in detail below and are accompanied by comments where needed.

```
/*
     This function initializes a linked list iterator, which consists of a pointer to a linked
     list and a pointer to the current link.  It sets the pointer to the linked list to the
     passed linked list. It sets the current link to the link that appears after the front
     sentinel.
*/
void linkedListIteratorInit(struct linkedList *lst, struct linkedListIterator * itr) {
   itr->lst = lst;
   itr->currentLink = lst->frontSentinel->next;
}
```

```
void _removeLink (struct linkedList * lst, struct dlink * lnk);

/*
     This function returns an integer value representing whether a linked list iterator has
     a next link.  If the linked list iterator does have a next link, then the function returns
     1.  Additionally, the function will set current to the next link in the linked list.
     If the iterator does not have a next link, then the function returns 0.
*/
int linkedListIteratorHasNext (struct linkedListIterator *itr) {
     if (itr->cur->next != itr->lst->backSentinel)
     {
          itr->cur = itr->cur->next;
          return (1);
     } else {
          return(0);
     }
}

/*
     This function returns the value at the next link. It does so by accessing the
     current link’s value and returning that value.
*/
TYPE linkedListIteratorNext (struct linkedListIterator *itr) {
     TYPE val = itr->cur->value;
     return val;
}


/*
     This function removes the value at the iterator’s current link. It does so by calling
     the _removeLink function we defined in a previous worksheet.  It also sets the value
     of the iterator’s current link to the link that came before the removed link prior to
     the removal.
```

```
*/
void linkedListIteratorRemove (struct linkedListIterator *itr) {
     struct DLink *tmp = itr->cur;
     itr->cur = itr->cur->prev;
     _removeLink(itr->lst, tmp);

}
```

Piazza Discussion

https://piazza.com/class/ib2kus4hsie528?cid=173