Subject: Boundary Testing
Posted by cdyancey on Wed, 25 Jan 2017 18:48:31 GMT
View Forum Message <> Reply to Message

I thought it would be a good idea for us to share our boundary tests. We’d like to think we have tested everything, but we don't know what we don't know!

I'll start:

- Test Union with an empty list
- Test removing all the items from a list, then adding again.
- Test DLL can display the items in reverse (this won't be tested but it could help ensure your links go both ways)

Subject: Re: Boundary Testing
Posted by cdyancey on Thu, 26 Jan 2017 03:55:57 GMT
View Forum Message <> Reply to Message

- Testing with values other than integers

Subject: Re: Boundary Testing
Posted by lusth on Thu, 26 Jan 2017 12:06:49 GMT
View Forum Message <> Reply to Message

This thread is a good idea.

Subject: Re: Boundary Testing
Posted by TitustheTitan on Thu, 26 Jan 2017 20:33:04 GMT
View Forum Message <> Reply to Message

How about using union to pass from one list to another, then back again?

Subject: Re: Boundary Testing
Posted by ncgleason on Thu, 26 Jan 2017 21:02:38 GMT
View Forum Message <> Reply to Message

I personally put a for loop running around a million times, using union to swap between two lists every time, to make sure it ran in constant time.
Subject: Re: Boundary Testing
Posted by bhpauken on Thu, 26 Jan 2017 22:38:57 GMT

- Printing sizes of both lists before and after Union

Subject: Re: Boundary Testing
Posted by ajherriott on Fri, 27 Jan 2017 18:30:15 GMT

Will there be a test for trying to union a list with itself? i.e. unionSLL(items, items);
Because right now, my union function ends with setting donor->head and donor->tail to NULL,
causing that function call to result in "items" being empty instead of [1,2,3,4,1,2,3,4] for example.

Subject: Re: Boundary Testing
Posted by luseth on Sat, 28 Jan 2017 14:56:08 GMT

^ That would be considered invalid input, but great outside-the-testing-dropbox thinking.