
linked_list Documentation

Release 0.0.1

Daniel Papp

February 21, 2017

1	Getting started with the linked_list module	1
1.1	Working with singly linked lists	1
1.2	Working with doubly linked lists	1
2	Operations on linked lists	3
2.1	Functions to add elements to the linked list	3
2.2	Functions to remove elements from a linked list	3
3	The node classes	5
4	Indices and tables	7

Getting started with the `linked_list` module

Working with singly linked lists

So for example you create a singly linked list(LL) with 5 elements in the following way using the `pushback` operation:

```
>>> import linked_list as ll
>>> lst = ll.LL(1)
>>> for i in xrange(2, 6):
...     ll.pushback(ll.LL(i))
```

This creates a linked list with 5 nodes and `lst` as the head element. Suppose now we want to pop the last element. We can do it with the `popback` operation like this:

```
>>> ll.popback(lst)
4
```

Now our list has only 4 elements. Now suppose we want to delete the third element. We can do this with the `delete` operation:

```
>>> ll.delete(lst, lst.next.next)
>>> lst.next.next.data
3
```

As we can see this deleted the third element from the list.

Working with doubly linked lists

All the operations of the singly linked lists also support doubly linked lists(DLL). Let's create a linked list by pushing elements with `pushfront` to the beginning of the list:

```
>>> import linked_list as ll
>>> lst = ll.DLL(4)
>>> for i in xrange(3, -1, -1):
...     ll.pushfront(lst, ll.DLL(i))
...     lst = lst.prev
```

So our list will have 5 elements just like in the singly linked list example but now it's a doubly linked list. The only operation that we haven't seen before is the `popfront` operation. Let's see an example for that one too:

```
>>> lst = lst.nxt
>>> ll.popfront(lst)
0
```

And basically that's all what this package is currently capable of.

Operations on linked lists

Functions to add elements to the linked list

Functions to remove elements from a linked list

The node classes

Indices and tables

- `genindex`
- `modindex`
- `search`