ECE 250 - Project 1
Dynamic Deques
Design Document
Julia Aziz, UW User ID: ja2aziz
Oct 18, 2022

Number of Pages: 2
Document name: ja2aziz_design_p1.pdf

1. Overview of classes

Class: Node

Description: Holds the information of each node. Each object of this class holds the URL name and the URL. These objects are used in the class "linkedlist" and "deque" to execute different commands requested by the user, such as adding, deleting, or finding URLs.

Member variables:

URL_name - stores the name of the URL, for example "google"
URL - stores the URL, for example "www.google.ca"
*prev - stores the pointer to the previous node, for example if the linked list is currently holding firefox->google->safari, prev of "google" would point to "firefox"
*next - stores the pointer to the next node, for example, if the linked list is currently holding firefox->google->safari, next of "google" would point to "safari"

Member functions:

Node - creates an object of the class Node and sets next and prev to nullptr and URL_name and URL to empty strings
Node - creates an object of the class Node and assigns the data given by the user to URL_name and URL
~Node - sets the pointers next and prev to nullptr
get_URL_name - returns the name of the URL since URL_name is a private member variable
get_URL - returns the URL since URL is a private member variable
set_URL_name - sets the name of the URL since URL_name is a private member variable
set_URL - sets the URL since URL is a private member variable

Class: linkedlist

Description: Fully implemented doubly linked list class that runs commands like insert, delete and print.

Member variables:

*head - an object of the class Node that is the head of the linked list
*tail - an object of the class Node that is the tail of the linked list
Curr_size - an integer that keeps track of how many URLs are added and removed

Member Functions:

linkedlist - sets the head and tail to nullptr
~linkedlist - deletes all the nodes in the linked list
insert_front - adds a node at the front of the linked list
insert_back - adds a node at the end of the linked list
delete_front - deletes the first node of the linked list
delete_back - deletes the last node of the linked list
print_list - prints all the nodes of the linked list

Class: deque

Description: This class inherits from the linkedlist class and executes user indicated commands such as pushing, popping and retrieving information

Member variables: This class has no member variables since it uses those from the linkedlist class

Member Functions:

deque - empty constructor - calls the linkedlist class constructor
push_front - calls insert_front from the linkedlist class to add a node at the front of the linked list and deletes the last node by calling delete_back if the linked list is full
push_back - calls insert_back from the linkedlist class to add a node at the back of the linked list and deletes the first node by calling delete_front if the linked list is full
pop_front - calls delete_front from the linkedlist class to delete the first node of the linked list
pop_back - calls delete_back from the linkedlist class to delete the last node of the linked list
clear - recursively calls delete on the nodes until curr_size is 0
size - returns the current size of the linked list
front - returns the first node of the linked list
back - returns the last node of the linked list
empty - checks and returns if the linked list is empty
find - iterates through the linked list to find the node that the user requested, returns found or not found