

Assignment- 8



Name:- Shaikryshna S Zirape

R.NO :- 21286.

Problem Statement:-

Second year Computer Engineering class, set A of students like Vanilla Ice-Cream and set B of students like Butterscotch ice-cream. Write C++ program to store two sets using linked list compute and display.

- set of student who like both vanilla & butterscotch
- set of student who likes either vanilla or butterscotch.
- NO of student who like neither vanilla nor butterscotch.

Objective:-

- To understand Concept of linked list.
- To study different set operation using set linked list.

Outcome:-

- To write a function of set using linked list.
- perform the respective set operation.

6/4 packages and hardware Components used.

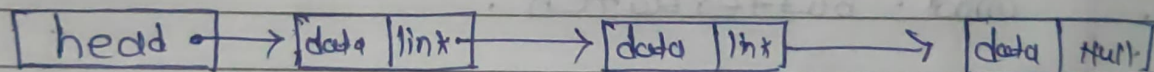
operating system (64 bit) Fedora 17.
programming tools (64 bit) IDEs: Open
Source update of Eclipse programming
Framework: C++.

1 theory:-

linked list:-

A linked list is a sequence of data structure which are connected through links.

Linked list is the sequence of links which contains items. Each link contains connection to another link.



Types of linked list:-

① Simple linked list:- Item Navigation in Forward way only.

② Doubly linked list:-

item navigated in Forward & Backward way.

③ Circular linked list:-

last item contains link of first element as next member

Basic Operations:-

Insertion:- add element at beginning.

Deletion:- Delete element at beginning.

Display:- Display Complete list.

Search:- Search an element using key.

Delete:- Delete element using key

* Algorithm:-

* addnode (char n)

// add node at beginning of list.

{

node *temp = null

t = new node;

t -> name = n

t -> next = head;

}

* displayelement ()

{

// display name of all student.

node *curr = head;

print("List of student")

while (curr != null) {

print (curr -> name);

print (curr -> next);

}

Intersection (linked list 1, linked list 2)

```
{
node * temp1 = head1
while (temp1 != NULL)
{
if (temp1->name == curr2->name) {
add node (curr2->name);
curr2 = curr2->next;
temp1 = temp1->next;
}
```

Difference (linked list 1, linked list 2)

```
node * curr1 = l1.head;
node * curr2 = l2.head;
while (curr1 != NULL) {
int count = 0;
while (curr2 != NULL) {
if (curr1->name == curr2->name) {
count++;
curr2 = curr2->next;
}
}
```

Complexity.

① Union.

space = $O(m+n)$

time = $O(m \times n)$

② Intersection

space = $O(\min(m, n))$

time = $O(m+n)$

③ Diff.

time = $O(m)$

space = $O(m)$

Test Case.	Description.	Output.	exp output.	Status.
①	$S = \{a, b, c, d, e, f, g\}$ $A = \{a, b, c, d, e\}$ $B = \{d, e, f\}$	d, e a, b, c, f g	d, e a, b, c, f g	pass.
②	$S = \{a, b, c, d\}$ $A = \{a, b\}$ $B = \{c, d\}$	$-$ a, b, c, d $-$	$-$ a, b, c, d $-$	pass.

Conclusion!

Successfully implemented the set operation using linked list.