

ΟΝΟΜΑ: Μηλιτιάδης
ΕΠΩΝΥΜΟ: Μαντές
ΑΜ: 1084661
ΕΤΟΣ: 2

10

ΑΣΚΗΣΗ ΕΡΓΑΣΤΗΡΙΟΥ : 4

■ ΣΧΟΛΙΑΣΜΟΣ ΛΥΣΗΣ

Ορίζουμε μια συνάρτηση `Search()` με ορίσματα έναν διπλό δείκτη `**head` τύπου `Node` που δείχνει στο πρώτο στοιχείο της λίστας και μια `int` μεταβλητή `a` που αντιπροσωπεύει την τιμή που θέλουμε να αναζητήσουμε στη λίστα. Με την συνάρτηση ελέγχουμε αν υπάρχει η τιμή αυτή και αν είναι πρώτη στην λίστα ή βρίσκεται ενδιάμεσα τότε γίνεται εναλλαγή θέσης με την δεξιότερη τιμή (`swap`) ή αν είναι τελευταία στην λίστα δεν γίνεται καμία εναλλαγή. Αρχικά, ορίζουμε για την αναζήτηση της τιμής `a` δύο δείκτες `*curA`, `*prevA` τύπου `Node` με τον πρώτο να δείχνει στο πρώτο στοιχείο της λίστας (`head`) και τον δεύτερο να είναι κενός (`NULL`). Με ένα βρόχο `while`, όσο η τιμή `a` που ψάχνουμε δεν είναι στο πρώτο κόμβο της λίστας θέτουμε τον `curA` να δείχνει στον αμέσως επόμενο κόμβο της λίστας και τον `prevA` να δείχνει στον τωρινό κόμβο. Δημιουργούμε δύο αντίστοιχους δείκτες `*curB`, `*prevB` που χρησιμοποιούνται για τον προσδιορισμό του ακριβώς δεξιότερου κόμβου, οπότε θέτουμε τον `curB` να δείχνει στον επόμενο κόμβο από αυτόν που δείχνει ο `curA`, δηλαδή στον μεθεπόμενο. Αν το περιεχόμενο του κόμβου στον οποίον δείχνει ο `curA` ισούται με `a` τότε εκτυπώνουμε ότι η τιμή βρέθηκε στη λίστα και ξεκινάμε τη διαδικασία του `swap`. Αν ο επόμενος κόμβος από αυτόν που δείχνει ο `curA` είναι κενός (`NULL`) τότε έχουμε φτάσει στον τελευταίο κόμβο της λίστας και εκτυπώνουμε ότι δεν πραγματοποιείται `swap`. Αν η τιμή `a` δεν είναι στον πρώτο κόμβο της λίστας θέτουμε τον `prevA` να δείχνει στον κόμβο που δείχνει και ο `curB`, αλλιώς ορίζουμε τον δεξιότερο κόμβο σαν `*head`. Όμοια, αν η τιμή στον δεξιότερο κόμβο δεν είναι στον πρώτο κόμβο της λίστας θέτουμε τον `prevB` να δείχνει στον κόμβο που δείχνει και ο `curA`, αλλιώς ορίζουμε τον κόμβο που περιέχει την τιμή `a` σαν `*head`. Τέλος, αν η τιμή `a` βρίσκεται στον πρώτο κόμβο ο δείκτης `*head` θα λάβει τη τιμή του `curB` και ο `prevB` θα δείχνει πλέον στον κόμβο που δείχνει και ο `curA`. Έπειτα, ορίζουμε έναν άλλο δείκτη `*temp` στον οποίο αποθηκεύουμε τον κόμβο στον οποίον δείχνει ο `curB` και θέτουμε τον `curB` να δείχνει στον κόμβο που δείχνει ο `curA` και τον `curA` να δείχνει στον κόμβο που δείχνει ο `temp`. Έτσι, εναλλάσσουμε τους δύο κόμβους χωρίς να εναλλάξουμε τις τιμές τους.

▪ ΚΩΔΙΚΑΣ ΛΥΣΗΣ

```
#include <iostream>
using namespace std;

struct Node
{
    int data;
    struct Node *prev;
    struct Node *next;
};

struct Node* head = NULL;          //Create new "node"
struct and a pointer named head

void put(int newdata)              //The function "put" creates
a new node and stores the new number
{
    struct Node* newnode = (struct Node*)
malloc(sizeof(struct Node)); //Allocates a "node"
structure size block of memory, returning a pointer to
the beginning of the block.
    (*newnode).data = newdata;
    (*newnode).prev = NULL;
    (*newnode).next = head;
    if(head != NULL)
        (*head).prev = newnode ;
    head = newnode;
}

void print()                       //Displays the list
{
    struct Node* ptr;
    ptr = head;
    while(ptr != NULL)
    {
        cout<< (*ptr).data <<" ";
        ptr = (*ptr).next;
    }
}

void Search(Node **head, int a)
{
    Node *curA = *head;
    Node *prevA = NULL;

    while(curA && curA->data != a)
    {
        prevA = curA;
        curA = curA->next;
    }
}
```

```

Node *curB = curA->next;
Node *prevB = curA;

    if (curA->data == a) cout<<endl<<"Number "<<a<<" is
found in the linked list.";
    if (curA == NULL || curB == NULL) cout<<endl<<"No
swap occurs because "<<a<<" is the last element of the
linked list.";
    if (prevA != NULL) prevA->next = curB;
    else *head = curB;

    if (prevB != NULL) prevB->next = curA;
    else *head = curA;

    if (prevA == NULL)
    {
        *head=curB;
        prevB->next = curA;
    }

Node *temp;
temp = curB->next;
curB->next = curA->next;
curA->next = temp;

    cout<<endl<<"The new linked list after the swap is :
";
    print();
}

int main()
{
    int num;

    put(17);
    put(62);
    put(5);
    put(34);
    put(22);
    put(46);

    cout<<"The stored numbers in the linked list are : ";
    print();

    cout<<endl<<"Please enter a number to be found in the
linked list : ";
    cin >> num;

    Search(&head, num);

```

```
    return 0;  
}
```

▪ **ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ**

a. Έστω ότι ψάχνουμε έναν αριθμό της λίστας εκτός από τον τελευταίο :

The stored numbers in the linked list are : 46 22 34 5 62
17

Please enter a number to be found in the linked list : 34

Number 34 is found in the linked list.

The new linked list after the swap is : 46 22 5 34 62 17

b. Έστω ότι ψάχνουμε τον τελευταίο αριθμό της λίστας :

The stored numbers in the linked list are : 46 22 34 5 62
17

Please enter a number to be found in the linked list : 17

Number 17 is found in the linked list.

No swap occurs because 17 is the last element of the linked list.