Indian Institute of Technology (Indian School of Mines), Dhanbad

Algorithm Design & Analysis Lab-Report

Submitted by:

Shubham Maurya
16JE002437
4th Sem., CSE

INDEX

LAB-3

Date:29/01/2018

S. No.	Question	Page	Remarks
1	Suppose that each row of an n × n array Arr consists of 1's and 0's such that, in any row of Arr, all the 1's come before any 0's in that row. Assuming Arr is already in memory, write a program running in O(n) time for finding the row of Arr that contains the most 1's.		
2	You are given two linked lists A and B which may or may not contain a common node. From the first common node (if any) in A and B, the two lists are the same until the end. The two lists in presence of a common node look like the Roman letter Y.		
3	You are given an n* n board. Your task is to place m coins on the board such that no two of the coins go to the same cell or to two adjacent cells. Two cells are called adjacent if their boundaries share an edge. Two cells with boundaries sharing only a corner will not be called adjacent. Write a program that prints all possible arrangements of m coins in the n*n board.		
4	You are provided an unsorted array A having size n that may or may not contain duplicates and a Number k (where,k<n). Your task is to write a program that runs in O(n) time and returns true if array A contains duplicates within a distance of k		

1.Suppose that each row of an n × n array Arr consists of 1's and 0's such that, in any row of Arr, all the 1's come before any 0's in that row. Assuming Arr is already in memory, write a program running in O(n) time for finding the row of Arr that contains the most 1's.

```
#include<bits/stdc++.h>
using namespace std;
int mem[100][100];
int a[200][200];
int dp(int i,int j){
        if(i==0||j==0)
                return 0;
        if(mem[i][j]!=0)
                return mem[i][j];
        if(a[i][j]==1)
                return mem[i][j]=1+dp(i,j-1);
        if(a[i][j]==0)
                return mem[i][j]=dp(i,j-1);
}
int main(){
        int n,t  ;
        cout<<"Enter matrix size ";
        cin>>n;
        cout<<"Enter the element"<<endl;
        for(int i=1;i<=n;i++){
                for(int j=1;j<=n;j++){
                        cin>>a[i][j];
                        mem[i][j]=0;
       }}
        mem[n][n]=0;
        for(int i=1;i<=n;i++)
                int ans = dp(i,n);
        int max=0,index;
        for(int i=1;i<=n;i++){
                if(mem[i][n]>max){
                        max=mem[i][n];
                        index=i;
                }
        }
        cout<<"Row that contain maximum 0 is :"<<index<<endl;
        return 0;
```

```
Enter matrix size 5
Enter the element
1 1 1 1 1
0 0 0 1 1
1 0 0 1 1
0 0 0 0 0
1 1 0 0 0
Row that contain maximum 0 is :1
```

}

2.You are given two linked lists A and B which may or may not contain a common node. From the first common node (if any) in A and B, the two lists are the same until the end. The two lists in presence of a common node look like the Roman letter Y.

```
#include<bits/stdc++.h>
using namespace std;

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

node* insert(node *s,int x)
{
	node *temp=new node();
	temp->next = NULL;
	temp->prev = NULL;
	temp->data=x;
	if(s==NULL)
		return temp;
	else
	{
		node *t=s;
		while(t->next!=NULL)
			t=t->next;
		t->next=temp;
		temp->prev=t;
		return s;
	}
}

void print(node *a,node *b)
{
	node *t1,*t2;
	t1=a;
	t2=b;
	if(t1==NULL && t2==NULL)
	{
		cout<<"Both lists are empty\n";
```

```
        return;
}
if(t1==NULL || t2==NULL)
{
        if(t2==NULL)
        {
                cout<<"List B is empty\n";
                cout<<"List A is:-\n";
                while(t1!=NULL)
                {
                        cout<<t1->data<<" ";
                        t1=t1->next;
                }
                cout<<"\n";
        }
        else
        {
                cout<<"List A is empty\n";
                cout<<"List B is:-\n";
                while(t2!=NULL)
                {
                        cout<<t2->data<<" ";
                        t2=t2->next;
                }
                cout<<"\n";
        }
        return;
}
while(t1->next!=NULL)
        t1=t1->next;
while(t2->next!=NULL)
        t2=t2->next;
if(t1!=t2)
{
        cout<<"Both list are not merged at all\n";
        t1=a;
        t2=b;
        cout<<"List A is:-\n";
        while(t1!=NULL)
        {
                cout<<t1->data<<" ";
                t1=t1->next;
        }
        cout<<"\n";
        cout<<"List B is:-\n";
```

```
        while(t2!=NULL)
        {
                cout<<t2->data<<" ";
                t2=t2->next;
        }
        cout<<"\n";
}
else
{
        t1=a;
        while(t1!=NULL)
        {
                t2=b;
                while(t2!=NULL)
                {
                        if(t1==t2)
                                break;
                        t2=t2->next;
                }
                if(t2!=NULL)
                        break;
                t1=t1->next;
        }
        cout<<"List A before merging is:-\n";
        t2=a;
        while(t2!=t1)
        {
                cout<<t2->data<<" ";
                t2=t2->next;
        }
        cout<<"\n";
        cout<<"List B before merging is:-\n";
        t2=b;
        while(t2!=t1)
        {
                cout<<t2->data<<" ";
                t2=t2->next;
        }
        cout<<"\n";
        cout<<"Merged list is:-\n";
        while(t1!=NULL)
        {
                cout<<t1->data<<" ";
                t1=t1->next;
        }
```

```
                cout<<"\n";
        }
}

int main()
{
        node *a=NULL,*b=NULL;
        int n,i,j,d,flag=0;
        pair <node*,node*> r;
        cout<<"Enter number of nodes you want to insert\n";
        cin>>n;
        cout<<"Enter 0 to merge two lists\n1 to insert in A\n2 to insert in B\n";
        for(i=0;i<n;i++)
        {
                if(flag==0)
                {
                        cout<<"Enter choice and and data for "<<i+1<<" node\n";
                        cin>>j>>d;
                        if(j==0)
                        {
                                flag=1;
                                node *t1=a,*t2=b;
                                node *temp=new node();
                                temp->data=d;
                                temp->next=NULL;
                                temp->prev=NULL;
                                if(t1==NULL)
                                        a=temp;
                                else
                                {
                                        while(t1->next!=NULL)
                                                t1=t1->next;
                                        t1->next=temp;
                                        temp->prev=t1;
                                }
                                if(t2==NULL)
                                        b=temp;
                                else
                                {
                                        while(t2->next!=NULL)
                                                t2=t2->next;
                                        t2->next=temp;
                                }
                        }
                        else if(j==1)
```

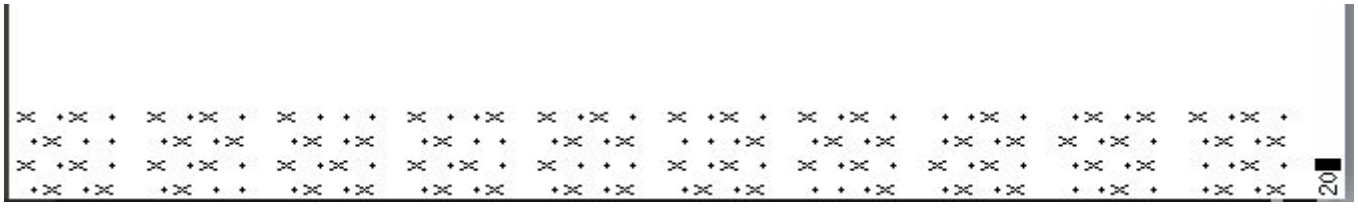
```
                        a=insert(a,d);
                else if(j==2)
                        b=insert(b,d);
            }
            else
            {
                cout<<"Enter only data for "<<i+1<<" node\n";
                cin>>d;
                a=insert(a,d);
            }
        }
        cout<<"\n";
        print(a,b);
        return 0;
}
```

```
bash
Enter number of nodes you want to insert
13
Enter 0 to merge two lists
1 to insert in A
2 to insert in B
Enter choice and and data for 1 node
1 7
Enter choice and and data for 2 node
1 5
Enter choice and and data for 3 node
2 3
Enter choice and and data for 4 node
2 9
Enter choice and and data for 5 node
1 6
Enter choice and and data for 6 node
0 13
Enter only data for 7 node
24
Enter only data for 8 node
13
Enter only data for 9 node
14
Enter only data for 10 node
10
Enter only data for 11 node
1
Enter only data for 12 node
9
Enter only data for 13 node
1

List A before merging is:-
7 5 6
List B before merging is:-
3 9
Merged list is:-
13 24 13 14 10 1 9 1
```

3.You are given an n* n board. Your task is to place m coins on the board such that no two of the coins go to the same cell or to two adjacent cells. Two cells are called adjacent if their boundaries share an edge. Two cells with boundaries sharing only a corner will not be called adjacent. Write a program that prints all possible arrangements of m coins in the n*n board.

```
#include<bits/stdc++.h>
using namespace std;

bool mat[100][100]={0};
int m,n;

bool possible(int i,int j){
	if(i-1>=0 && mat[i-1][j]==true)   //Top
		return false;

	if(i+1>=0 && mat[i+1][j]==true)   //Down
		return false;

	if(j-1>=0 && mat[i][j-1]==true)   //Left
		return false;

	if(j+1>=0 && mat[i][j+1]==true)   //Right
		return false;

	return true;
}

int ways(int i,int j,int coins)
{
	if(coins==0){
		for(int i=0 ; i<m ; i++){
			for(int j=0 ; j<n ; j++){
				if(mat[i][j])
					cout<<"X ";
				else
					cout<<". ";
			}
			cout<<endl;
		}
		cout<<endl;
```

```
                return 1;
        }
        if(i==m)
                return 0;
        if(j==n)
                return ways(i+1,0,coins);

        int ans=0;
        //If it is possible to place the queen at that place then place it
        if(possible(i,j)){
                mat[i][j]=1;
                ans+=ways(i,j+1,coins-1);
                mat[i][j]=0;
        }

        ans+=ways(i,j+1,coins);
        return ans;
}

int main()
{
        int coins;
        cin>>n;
        m=n;
        cin>>coins;
        cout<<"Total Ways  "<<ways(0,0,coins);
        return 0;
}
```



4.You are provided an unsorted array A having size n that may or may not contain duplicates and a Number k (where,k<n). Your task is to write a program that runs in O(n) time and returns true if array A contains duplicates within a distance of k.

```
#include<bits/stdc++.h>
using namespace std;

int a[1000],b[100000];

int main(){
	bool ans = false;
	int n,k,i,x;
	cout<<"Enter number of elements in the array and K : ";
	cin>>n>>k;
	cout<<"Now enter the elements of the array"<<endl;
	for(i=0;i<n;i++){
		cin>>x;
		a[x]++;
		if(a[x]>1)
			ans=true;
		b[i]=x;
	}
	for(;i<n;i++){
		a[b[i-k]]--;
		cin>>x;
		a[x]++;
		if(a[x]>1)
			ans=true;
		b[i]=x;
	}

	if(ans)
		cout<<"Duplicate found in the range of "<<k<<endl;
	else
		cout<<"No duplicate found in the range of "<<k<<endl;
	return 0;
}
```

```
Enter number of elements in the array and K : 15 5
Now enter the elements of the array
1 7 10 34 12 14 4 6 7 14 7 1 8 3 5
Duplicate found in the range of 5
```