**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Н-Н ІНСТИТУТ ПІДПРИЄМНИЦТВА ТА ПЕРСПЕКТИВНИХ ТЕХНОЛОГІЙ**



**ЗВІТ**

до виконаної лабораторної робіти №4

на тему

***«***Обхід графів***»***

Виконав:

студент гр. КН-214

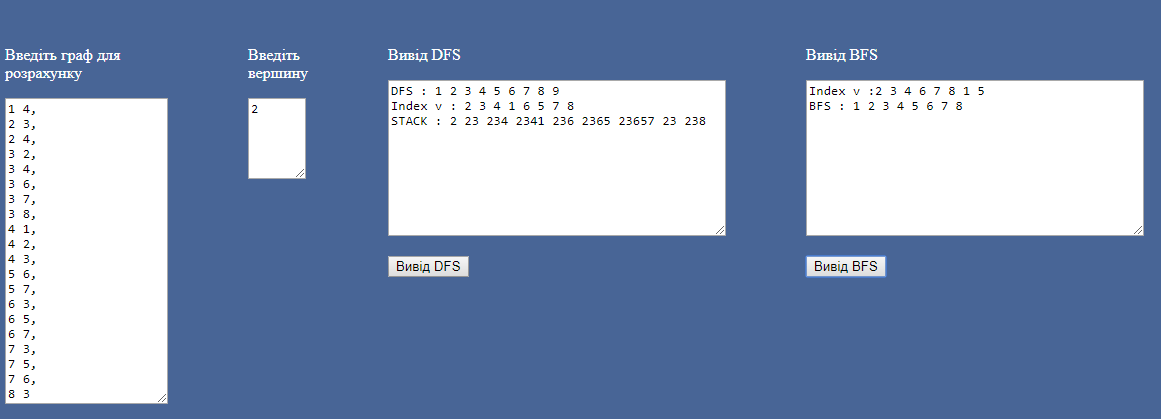
**Дубницький Ю. І.**

Прийняв:

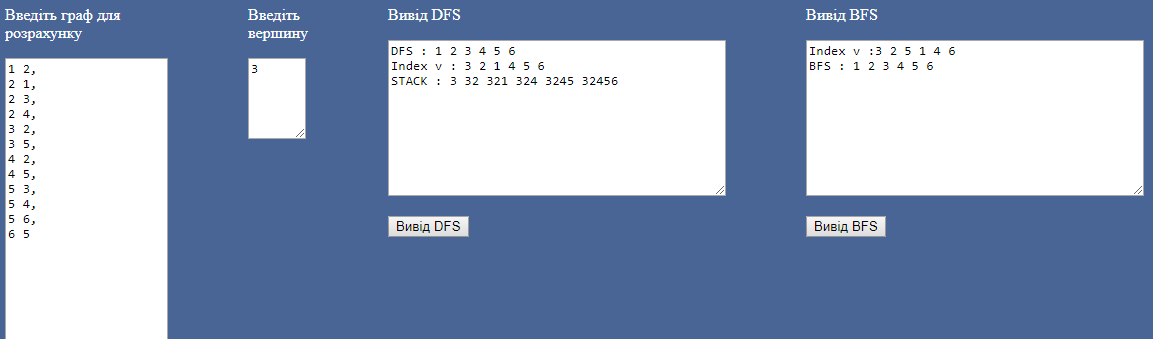
**Слюсарчук Ю.М.**

**Львів–2019**

**Тема:** Дослідити роботу алгоритмів обходу в графах: пошуку вшир та пошуку вглиб.







Код.

**function** start() {  
 **var** text = document.getElementById('start').value;  
 **function** starting( text){  
 **var** array = text.split(",");  
 **var** number = [];  
 **var** a;  
 **for**(**var** i =0;i<array.length;i++){  
 a = [array[i].trim().split(" ")];  
 number[i] = a;  
 }  
 **return** number;  
 }  
 **var** number = starting(text);  
 **return** number  
}  
/\*  
start()   
функція зчитує граф і записує в масив  
\*/  
**var** degvplus =[];  
**function** dfs() {  
 **var** ver = document.getElementById('ver').value;  
 **var** number = start();  
 **for** (**var** i = 1; i <= number.length; i++) {  
 **var** degplus = 0;  
 **for** (**var** j = 0; j < number.length; j++) {  
 **if** (number[j][0][0] == i) {  
 degplus++  
 }  
 }  
 degvplus.push(degplus)  
 }  
 /\*  
 записуєм степені графу  
 і викликаєм функцію deathFirstSearch()  
 \*/  
 deathFirstSearch(number,ver,degvplus)  
  
}  
  
  
**function** search(array,ver,vizut){  
 **for**(**var** i = 0;i<array.length;i++){  
 **if**(array[i][0][1]==ver){  
 vizut[array[i][0][0]-1]= vizut[array[i][0][0]-1]-1  
 }  
 }  
 **for**(**var** i =0 ;i<vizut.length;i++){  
 **if**(vizut[i]<0){  
 vizut[i]=0  
 }  
 }  
}  
/\*  
search()   
при виклику функції вона віднімає звязки задоної вершини   
\*/  
**function** newVersearch(numb,vizut) {  
 **for**(**var** i =0;i<vizut.length;i++){  
 **if**(vizut[i]==0){  
  
 }**else**{  
 **return** numb=i+1  
 }  
 }  
}  
/\*  
newVersearch()  
функція перевіряє чи є у вершини звязки  
\*/  
**function** output(text,array,values){  
 text.value+=values;  
 **for**(**var** i = 0;i<array.length;i++){  
 text.value+=array[i]+' '  
 }  
 text.value+='\n'  
}  
/\*  
output()  
функція виводить результати на екран  
\*/  
  
**function** deathFirstSearch(array,ver,vizut){  
 **var** varIndexClone=[];  
 **var** varIndex=[];  
 **var** verIndexNumb=ver;  
 **var** DFS = [];  
 **var** STACK=[];  
 **var** indexDfs = 1;  
 **var** verhun;  
 **var** newVer;  
 verhun = ver;  
 **var** str=''  
 **var** chexk=0;  
 **for** ( **var** i = 0;i<array.length;i++) {  
 **if** (!varIndex.includes(verIndexNumb+'')){  
 varIndex.push(verIndexNumb);  
 varIndexClone.push(verIndexNumb)  
 }  
 DFS.push(indexDfs);  
  
 **if**(i == 0 ){  
 STACK.push(verIndexNumb)  
 }**else**{  
 **for**(**var** i=0;i<varIndexClone.length;i++){  
 str += varIndexClone[i]+''  
 }  
 STACK.push(str)  
 str=''  
 }  
  
 **for**(**var** a = 0;a<vizut.length;a++) {  
 **if**(array[a][0][0] == verIndexNumb) {  
 **if**(verhun == verIndexNumb&&chexk==0){  
 search(array,verIndexNumb,vizut)  
 chexk++  
 }  
 **if**(varIndex.includes(array[a][0][1])){  
 **if**(vizut[verIndexNumb-1] == 0){  
 verIndexNumb= newVersearch(verIndexNumb,vizut);  
 newVer = verIndexNumb;  
 **for**(**var** i =varIndexClone.length;i>=varIndexClone.length;){  
 **if**(varIndexClone[--i]!=verIndexNumb){  
 varIndexClone.pop()  
 }  
 }  
 }  
 **continue**;  
 }**else**{  
 newVer = array[a][0][1];  
 **break**;  
 }  
 }  
 }  
 verIndexNumb = newVer;  
 verhun=newVer;  
 chexk=0;  
 indexDfs++;  
  
 **if**(varIndex.length==array[array.length-1][0][0]){  
 **var** text = document.getElementById('dfs');  
 output(text,DFS,'DFS : ')  
 output(text,varIndex,'Index v : ')  
 output(text,STACK,'STACK : ')  
 **return** }  
 }  
  
}  
/\*  
deathFirstSearch()  
функція виводить dfs  
\*/  
**function** bfs() {  
 **var** ver = document.getElementById('ver').value;  
 **var** number = start();  
 BFirstSearch(number,ver)  
  
}  
**function** BFirstSearch(array,ver) {  
 **var** verIndex = [];  
 **var** BFS=[];  
 **var** Guene=[];  
 **var** index = 1;  
 verIndex.push(ver);  
 BFS.push(index);  
 Guene.push(ver);  
 **var** numbIndex =0  
 **var** numb = verIndex[numbIndex];  
 **var** str=''  
 **var** last;  
 **var** someQueue;  
 **var** indexgo =0  
 **for**(**var** i =0;i<array.length;i++){  
 **if**(array[i][0][0]==numb){  
 **if**(verIndex.includes(array[i][0][1])){  
 **continue** }**else**{  
 **for**(**var** a = 0;a<array.length;a++){  
 **if**(array[a][0][0]==numb && !verIndex.includes(array[a][0][1])){  
 verIndex.push(array[a][0][1]);  
 BFS.push(++index)  
 }  
 }  
 numb=verIndex[++indexgo]  
 i=-1  
 a=0  
 }  
 }  
 **if**(9==BFS.length){  
 **break** }  
 }  
 **var** text=document.getElementById('bfs')  
 output(text,verIndex,'Index v :')  
 output(text,BFS,'BFS : ')  
}

Висновок: я навчився обчислювати алгоритми пошуку в глиб і ширину.