Лабораторна робота №6

Тема: Алгоритми пошуку в графі.

1. Обрав Обхід графа: пошук в глибину, пошук в ширину.

Псевдокод обох алгоритмів схожий, тільки в глибину використовується стек, а в ширину черга:

Нехай *G* – зв'язний граф. Треба, пересуваючись тільки по ребрам графа, обійти та  
розмітити всі його вершини.  
*Вхід*: граф *G*; *s*∈*V -* початок обходу  
*Вихід*: маркери вершин *v*.*num*  
*Допоміжні величини*:

|  |  |
| --- | --- |
| *T* ⊆ *V  v*.*selected  v*.*prev* | множина вершин, які потрібно розглянути чи вершина розглянута попередня вершина під час обходу |

1) для всіх *v*∈*V v*.*selected=false*; *v*.*prev=null*; *N=*1  
2) включити *s* у *T*; *s*.*selected=true*; *w*.*num=N*3) поки *T*≠∅ виконати  
4) видалити *u* з *T* /спосіб вибору *u* може бути різним  
5) для всіх *w*∈*Г*(*u*) виконати /для всіх *w* суміжних з *u*6) якщо *w*.*selected=false* то  
7) включити *w* у *T*8) *N=N*+1; *w*.*num=N*;  
9) *w*.*selected=true*10) *w*.*prev=u*

**Алгори́тм пошуку́ в глибину́** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *Depth-first search, DFS*) — [алгоритм](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) для обходу [дерева](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)), структури подібної до дерева, або [графа](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). Робота алгоритму починається з кореня дерева (або іншої обраної вершини в графі) і здійснюється обхід в максимально можливу глибину до переходу на наступну вершину.[[1]](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA_%D1%83_%D0%B3%D0%BB%D0%B8%D0%B1%D0%B8%D0%BD%D1%83#cite_note-mit-1)

**По́шук у ширину́** — [алгоритм](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) пошуку на [графі](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)).[[1]](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA_%D1%83_%D1%88%D0%B8%D1%80%D0%B8%D0%BD%D1%83#cite_note-mit-1)

Якщо задано граф *G* = (*V*, *E*) та початкову вершину *s*, алгоритм пошуку в ширину систематично обходить всі досяжні із *s* вершини. На першому кроці вершина *s* позначається, як пройдена, а в список додаються всі вершини, досяжні з *s* без відвідування проміжних вершин. На кожному наступному кроці всі поточні вершини списку відмічаються, як пройдені, а новий список формується із вершин, котрі є ще не пройденими сусідами поточних вершин списку. Для реалізації списку вершин найчастіше використовується [черга](https://uk.wikipedia.org/wiki/%D0%A7%D0%B5%D1%80%D0%B3%D0%B0_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)) та принцип [FIFO](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%B7%D0%B0%D0%BC%D1%96%D1%89%D0%B5%D0%BD%D0%BD%D1%8F_%D0%BA%D0%BE%D0%BC%D1%96%D1%80%D0%BE%D0%BA_%D0%BF%D0%B0%D0%BC%E2%80%99%D1%8F%D1%82%D1%96_FIFO). Виконання алгоритму продовжується до досягнення шуканої вершини або до того часу, коли на певному кроці в список не включається жодна вершина. Другий випадок означає, що всі вершини, доступні з початкової, уже відмічені, як пройдені, а шлях до цільової вершини не знайдений.

Клас складності обох алгоритмів O(|V|), де |V| - кількість вершин графа. Бо щоб знайти вершину, в найгіршому випадку треба перебрати всі вершини.

Прога:

lab\_6\_poshuk\_v\_glibinu\_i\_v\_sirinu\lab\_6\_poshuk\_v\_glibinu\_i\_v\_sirinu.

прога запускає цикли для різних даних і зберігає час виконання в файлі

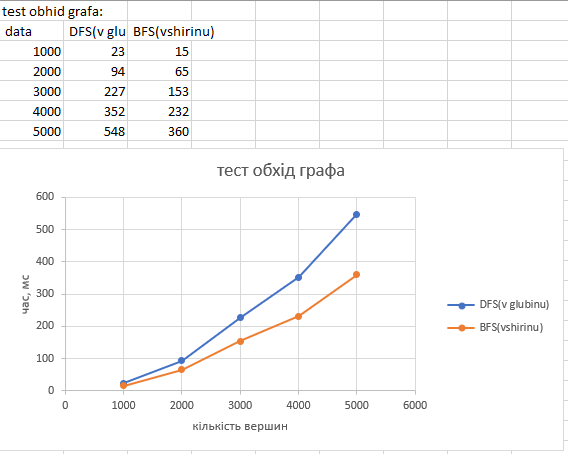
lab\_6\_poshuk\_v\_glibinu\_i\_v\_sirinu\bin\Debug\data1.csv

графіки є в файлі data6.xlsx

Файл для тестування алгоритмів з невеликою кількістю даних: \bin\Debug\data2.csv

Його вміст відповідає прикладу з файлу 6\_2 Обхід вершин.pdf

Графіки з програми схожі на лінійні.



Висновок: Обхід графа: пошук в глибину, пошук в ширину працюють з лінійною складністю, складність алгоритму O(|V|), де |V| - кількість вершин графа.

