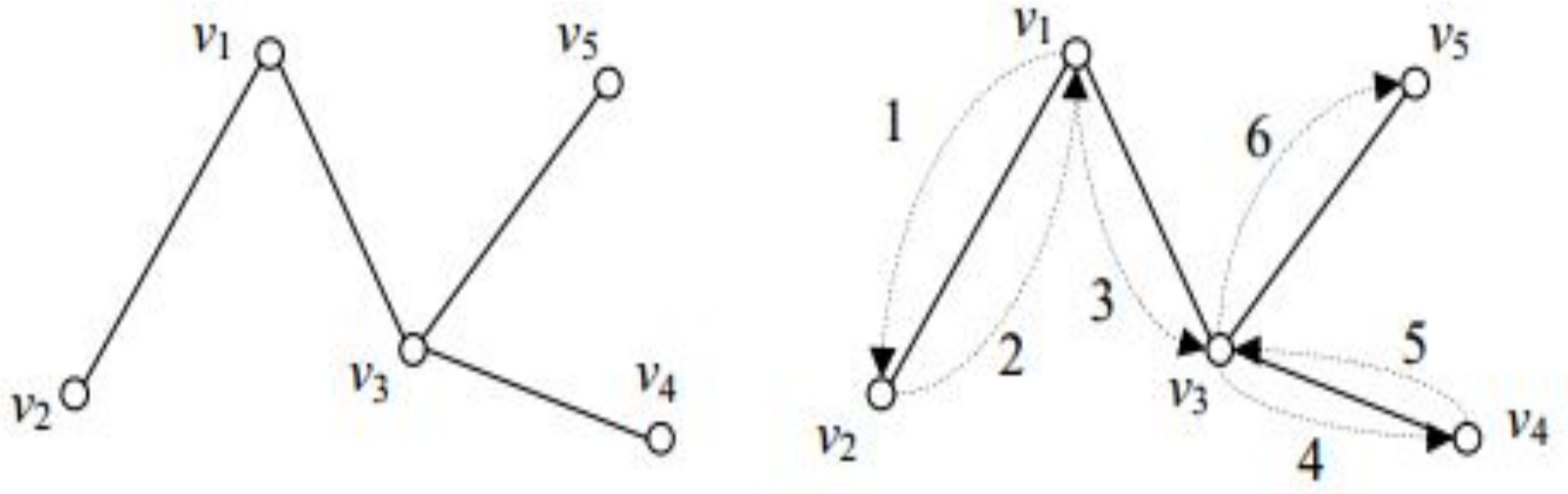
“Алгоритми пошуку оптимального рішення задачі та найкоротшого шляху”

Підготував:
студент групи ФЕІ-41
Коблан Ігор

Пошук маршруту у графі

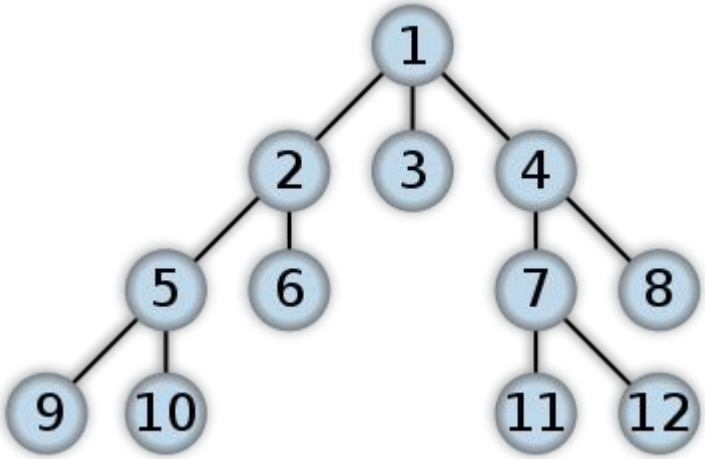
Алгоритм Террі

Знаходить шлях між двома вершинами графа

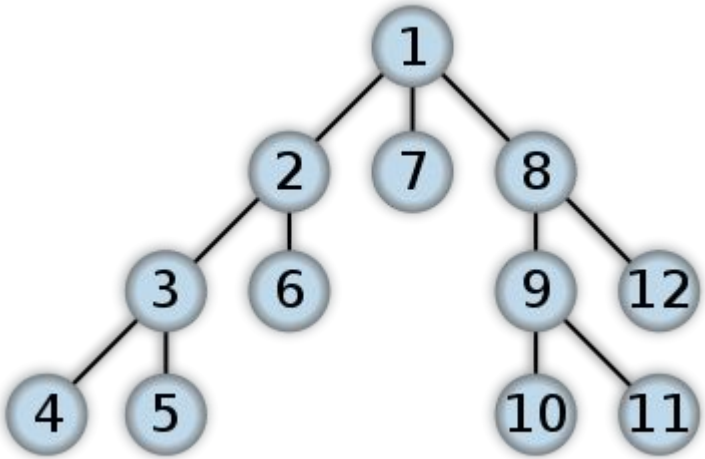


Обхід графів

Пошук у ширину



Пошук у глибину



Пошук відстані між вершинами графа

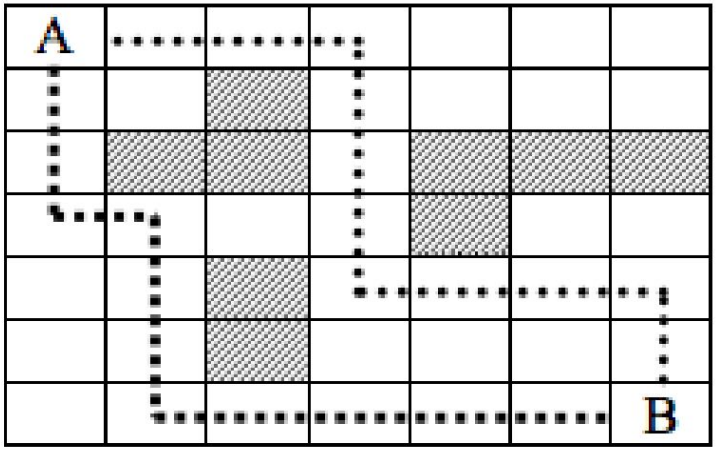
Хвильовий алгоритм

Хвильовий алгоритм - алгоритм, що дозволяє знайти мінімальний шлях в графі з ребрами одиничної довжини. Заснований на алгоритмі пошуку в ширину. Застосовується для знаходження найкоротшого шляху в графі, в загальному випадку знаходить лише його довжину.

Приклад роботи Хвильового алгоритму

0	1	2	3	4	5	6
1	2		4	5	6	7
2			5			
3	4	5	6		10	11
4	5		7	8	9	10
5	6		8	9	10	11
6	7	8	9	10	11	12

а)



б)

Приклад роботи двонапрямленого Хвильового алгоритму

0	1	2	3	4	5	6
1	2		4	5	6	
2			5			
3	4	5	6/6’		4’	3’
4	5		5’	4’	3’	2’
5	6/6’		4’	3’	2’	1’
6/6’	5’	4’	3’	2’	1’	0’

Алгоритм Дейкстри

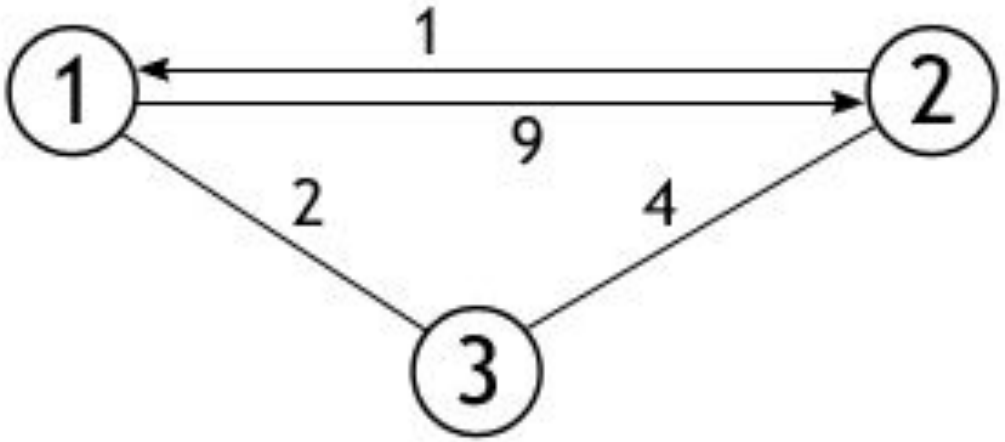
Алгоритм на графах, який знаходить найкоротший шлях від однієї вершини графа до всіх інших вершин. Класичний алгоритм Дейкстри працює тільки для графів без циклів від'ємної довжини.

Алгоритм Белмана — Форда

Алгоритм шукає найкоротший шлях від заданої вершини до всіх інших. Ідея в тому, що ми зберігаємо в масиві відстані до кожної вершини і на кожному кроці алгоритму оновлємо ці значення, якщо знайшли ребро яке покращує результат. Таким чином, після після кожного кроку матимемо масив найкоротших шляхів.

Алгоритм Флойда-Уоршола

Для k от 1 до |V| виконати
Для i от 1 до |V| виконати
Для j от 1 до |V| виконати
Якщо вага i->k и k->j не нульові, и i не рівно j, то
Якщо вага i->k + вес k->j меньше веса i->j, то заменить его их суммой



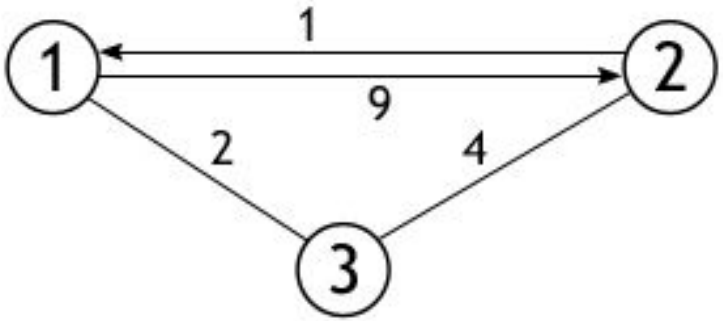
0	9	2
1	0	4
2	4	0

Матриця суміжності

k	i	j	ЗАМІНА
1	1	1	
1	1	2	
1	1	3	
1	2	1	
1	2	2	
1	2	3	3<4, D[2][3] ←3
1	3	1	
1	3	2	
1	3	3	
2	1	1	
2	1	2	
2	1	3	
2	2	1	
2	2	2	
2	2	3	
2	3	1	
2	3	2	
2	3	3	
3	1	1	
3	1	2	6<9, D[1][2] ←6
3	1	3	
3	2	1	
3	2	2	
3	2	3	
3	3	1	
3	3	2	
3	3	3	

0	9	2
1	0	4
2	4	0

0	9	2
1	0	3
2	4	0



0	6	2
1	0	4
2	4	0

Алгоритм Джонсона

Алгоритм Джонсона дозволяє знайти найкоротші шляхи між усіма парами вершин зваженого орієнтованого графу. Цей алгоритм працює, якщо у графі містяться ребра з додатними чи від’ємними вагами, але відсутні негативні цикли.

Висновки

Підіб'ємо підсумки. Якщо необхідно знайти відстань від однієї вершини до іншої або до всіх вершин графу і ваги всіх ребер графу є додатними або дорівнюють нулю, то найбільш ефективним виявляється алгоритм Дейкстри із часом роботи O(m lgn). Якщо ж ваги ребер можуть бути від'ємними, то необхідно застосовувати алгоритм Белмана-Форда, час роботи якого O(nm).
Якщо необхідно знайти відстані між усіма парами вершин графу, граф є розрядженим і всі ребра мають невід'ємні ваги, то можна виконати n разів алгоритм Дейкстри. Якщо ж граф є розрядженим, але в ньому можуть бути ребра з від'ємними вагами, то необхідно використовувати алгоритм Джонсона. Якщо необхідно знайти відстані між усіма парами вершин, ваги ребер можуть бути від'ємними і граф не є розрядженим (m прямує до n^2), то необхідно використовувати алгоритм Флойда-Уоршола.
Жоден з наведених алгоритмів не може бути застосований для графів, які містять негативні цикли. Проте алгоритм Белмана-Форда (як і алгоритм Джонсона), а також алгоритм Флойда-Уоршола можуть виявити такі цикли.

Дякую за увагу