

Санкт-Петербургский Политехнический Университет
Высшая школа прикладной математики и вычислительной физики, ФизМех
01.03.02 Прикладная математика и информатика

Лабораторная работа №4

Дисциплина “Дискретная математика”

Тема “ Циклы и раскраска”

Вариант “ Наибольшее независимое множество
вершин и наименьшее вершинное покрытие”

Поставленная задача

Найти в графе и вывести любые наибольшее независимое множество вершин и наименьшее вершинное покрытие.

Используемый язык программирования

Python 3.12.6

Описание алгоритма

Нахождение наибольшего независимого множества вершин:

Инициализация:

$X \leftarrow \emptyset$

$\text{max_size} \leftarrow 0$

$\text{stack} \leftarrow [(\emptyset, \{0, 1, 2, \dots, V-1\})]$

Пока stack не пуст:

$(S, T) \leftarrow \text{stack.pop}()$

Если $|S| > \text{max_size}$:

$X \leftarrow S$

$\text{max_size} \leftarrow |S|$

Для каждой вершины v из T :

Если $\forall u \in S: \text{adj_matrix}[v][u] = 0$:

$\text{new_T} \leftarrow T \setminus \{v\}$

$\text{stack.push}((S \cup \{v\}, \text{new_T}))$

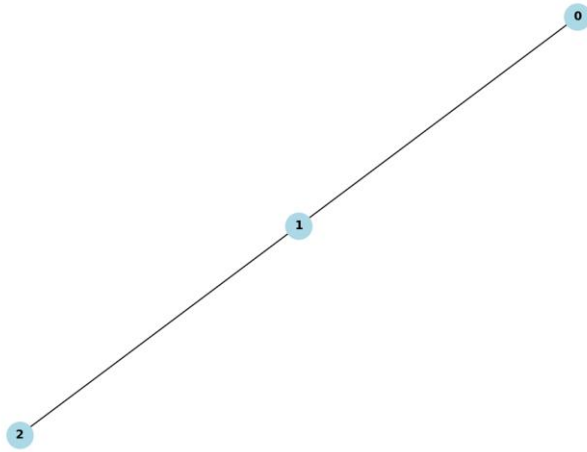
Вернуть X

Нахождение наименьшего вершинного покрытия множества:

$\text{min_vertex_cover} = [v \text{ for } v \text{ in range(self.V) if } v \text{ not in max_independent_set}]$

Пример работы алгоритма

Для примера рассмотрим граф.



	0	1	2
0	0	1	0
1	1	0	1
2	0	1	0

Поиск наибольшего независимого множества:

- Инициализация:

- Стек: $[([], [0, 1, 2])]$ (пустое множество и все вершины в списке T).

- Итерации:

- Взять из стека пару $S = [], T = [0, 1, 2]$.

- Добавить вершину 0 в S: $S = [0], T = [1, 2]$.

- Проверить вершину 1: она связана с 0, пропустить.

- Проверить вершину 2: она не связана с 0, добавить: $S = [0, 2]$.

- Результат: Наибольшее независимое множество $S = [0, 2]$.

Поиск наименьшего вершинного покрытия:

- Вершины графа: $V = \{0, 1, 2\}$.

- Независимое множество: $I = \{0, 2\}$.

- Наименьшее вершинное покрытие:

$\min_vertex_cover = V \setminus I = \{1\}$.

Сложность алгоритма

Алгоритм имеет экспоненциальную сложность $O(2^V \cdot V)$. Общее число таких подмножеств — 2^V . В худшем случае алгоритм выполняет $O(2^V)$ итераций. Наименьшее вершинное покрытие можно получить, исключив вершины наибольшего независимого множества из всех вершин графа, выполняется за $O(V)$.

Входные и выходные данные

Входные данные. Квадратная матрица $n \times n$, где n — количество вершин в графе. Каждая строка матрицы представляет связи (ребра) для одной вершины.

Выходные данные. Записывается в файл строки вида:

Наибольшее независимое множество вершин: {наибольшее независимое множество вершин}

Наименьшее вершинное покрытие: {множество вершин которое является наименьшим вершинным покрытием}

Область применимости

Алгоритм для нахождения максимального независимого множества (MIS) и минимального вершинного покрытия (MVC) подходит для определённых типов задач и графов. MIS часто используется для разделения графов на независимые компоненты или кластеры. Пример — задачи кластеризации в сетях, когда нужно выделить группы объектов, которые не взаимодействуют напрямую. MVC используется для минимизации количества узлов, которые должны быть подключены для обеспечения полной связи между всеми элементами сети. В задачах настройки беспроводных и проводных сетей — это критично для сокращения затрат на оборудование и обеспечение эффективной работы сети.

Представление графов в программе

Для представления графа в программе я буду использовать матрицу смежности. Доступ к данным в матрице занимает $O(1)$ что делает возможным прямую и быструю работу с каждой парой вершин. Кроме того, добавление или удаление ребра также выполняется за $O(1)$, так как достаточно изменить один элемент матрицы.

Вывод

Алгоритмы для нахождения максимального независимого множества (MIS) и минимального вершинного покрытия (MVC) являются важными инструментами в теории графов и находят широкое применение в различных областях, таких как оптимизация сетевых структур, планирование, биоинформатика и логистика. Несмотря на их теоретическую значимость, выбранный алгоритм имеет экспоненциальную сложность.