

Лабораторная работа № 9 по курсу Дискретный Анализ. Графы

Выполнил студент группы М8О-307Б-21 МАИ *Друхольский Александр*.

Условие

Задача:

Задан неориентированный двудольный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n . Необходимо найти максимальное паросочетание в графе алгоритмом Куна. Для обеспечения однозначности ответа списки смежности графа следует предварительно отсортировать. Граф не содержит петель и кратных ребер.

Метод решения

Для решения задачи поиска максимального паросочетания воспользуемся алгоритмом Куна. Для этого введём понятия чередующейся и увеличивающей цепи относительно паросочетания. Чередующаяся цепь - последовательность ребер, которые поочередно принадлежат и не принадлежат паросочетанию. Увеличивающая цепь - чередующаяся цепь, начальное и конечное ребро которой не принадлежат паросочетанию. Для построения алгоритма воспользуемся двумя правилами:

1. Паросочетание является максимальным, если в графе нет увеличивающей цепи.
2. Если из вершины не существует увеличивающей цепи в текущем паросочетании, то и в последующих паросочетаниях, которые можно достичь из текущего, для этой вершины не будет увеличивающей цепи.

Для реализации алгоритма сначала разделим граф на две доли (осуществим это с помощью set'ов). Далее нам нужно выбрать одну из долей. Поочередно для каждой вершины из выбранной доли запускаем измененный поиск в глубину, и если вершина, в которую перешли, ещё не принадлежит паросочетанию, то добавим её к паросочетанию и перейдём к следующей вершине выбранной доли. Если же она уже принадлежит паросочетанию, то запускаем поиск увеличивающей цепи уже для вершины, с которой она соединена (пока не встретим вершину, которую уже посещали на данной итерации). Если увеличивающая цепь найдена, то переназначаем пары текущего паросочетания. Когда все вершины одной доли будут рассмотрены, текущее паросочетание будет максимальным. Сложность алгоритма $O(nm)$, где n - количество вершин в выбранной доле, а m - количество ребер. Сложность получится таковой, так как для всех вершин выбранной доли нам нужно запустить поиск в глубину.

Описание программы

Основные моменты:

1. *int Kuhn(...)* - непосредственно сам алгоритм, который является по сути изменённым поиском в глубину
2. В *main* с помощью сетов разделяем входные данные на левую и правую долю, а затем запускаем алгоритм поиска в глубину для каждой вершины доли. Вектора *pr* - для паросочетаний, *visited* - для проверки посещения вершины, *right* - вершины из выбранной доли. Для сортировки вывода создан вектор *ges* и написана функцию сравнения (*cmp*).

Дневник отладки

RE - 6 Нарушение работы алгоритма, так как не были учтены особенности ввода (граф неявно двудольный)

WA - 4, 6, 7 Ошибка в логике разбиения графа на доли. Стал применять *unordered set* для разбиения входных данных.

Тест производительности

Исследуем сложность $O(nm)$

№	N	M	Время, с
1	100	70	0.000027
2	1000	700	0.001484
3	10000	7000	0.062171
4	100000	70000	8.53552

Данные будем увеличивать в 10 раз для наглядности. Как мы видим, с небольшой погрешностью, время выполнения как и входные данные увеличивается примерно в 100 раз.

Выводы

Я реализовал алгоритм Кунна поиска максимальных паросочетаний. Улучшил свои навыки в работе с графами и придумал алгоритм разбиения графа на доли, если он задан в неявном виде. Мне удалось реализовать алгоритм в соответствии с установленной сложностью алгоритма Кунна.