Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №9 по курсу «Дискретный анализ»

Студент:	А. О. Ларченко
Преподаватель:	А. А. Кухтичев
Группа:	М8О-206Б
Дата:	
Оценка:	
Подпись:	

Москва, 2025

Лабораторная работа №9

F. 8 Поиск максимального паросочетания алгоритмом Куна

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.

Формат ввода

В первой строке заданы 1<=n<=110000 и 1<=m<=40000. В следующих m строках записаны ребра. Каждая строка содержит пару чисел – номера вершин, соединенных ребром.

Формат вывода

В первой строке следует вывести число ребер в найденном паросочетании. В следующих строках нужно вывести сами ребра, по одному в строке. Каждое ребро представляется парой чисел – номерами соответствующих вершин. Строки должны быть отсортированы по минимальному номеру вершины на ребре. Пары чисел в одной строке также должны быть отсортированы.

1 Описание

Чтобы преступить к разбору алгоритма Куна, следует начасть с определения основных терминов, которые используются в данном алгоритме.

Паросочетание (matching) M в двудольном графе — произвольное множество рёбер двудольного графа такое, что никакие два ребра не имеют общей вершины.
Максимальное паросочетание — это такое паросочетание M в графе G, которое не содержится ни в каком другом паросочетании этого графа, то есть к нему невозможно добавить ни одно ребро, которое бы являлось несмежным ко всем рёбрам паросочетания.
Полное паросочетание - это такое паросочетание M в графе G, которое окрывает все вершины графа.
Чередующаяся цепь (alternating path) — путь в двудольном графе, для любых двух соседних рёбер которого верно, что одно из них принадлежит паросочетанию M, а другое нет.
Дополняющая(увеличивающая) цепь — чередующаяся цепь, у которой оба конца свободны.[1]

Алгоритм: Вначале берем пустое паросочетание, а потом — пока в графе удаётся найти увеличивающую цепь, — будем выполнять чередование паросочетания вдоль этой цепи, и повторять процесс поиска увеличивающей цепи. Как только такую цепь найти не удалось — процесс останавливаем, — текущее паросочетание и есть максимальное. Данное утверждение справедливо в следствии теоремы Бержа:
"Если из вершины x не существует дополняющей цепи относительно паросочетания M и паросочетание M получается из M изменением вдоль дополняющей цепи, тогда из x не существует дополняющей цепи в M".[2]

Оценка сложности.
Мы запускаем обход в глубину из n вершин, сложность обхода в глубину - $O(n+m)$, следовательно сложность алгоритма Куна - $O(n(n+m))$.

2 Исходный код

В основном теле программы мы считываем граф, сортируем его вершины, а затем запускаем Алгоритм Куна из еще не обработанных вершин, на каждой его итерации обнуляем массив посещенных вершнн.

```
int main(){
    std::ios::sync_with_stdio(false);
    std::cin.tie(nullptr);
    int n, m;
    cin>>n>>m;
    vector<vector<int>> g(n+1);
    vector<int> matching(n+1, -1);
    int match_cnt = 0;
    for(int i =0; i<m;++i){
        int u, v;
        cin>>u>>v;
        g[u].push_back(v);
        g[v].push_back(u);
    }
    for(int i=1; i<n+1;++i){
        sort(g[i].begin(), g[i].end());
    }

    for(int i=1; i<n+1;++i){
        vector<int> used(n+1, 0);
        if (matching[i]==-1) dfs( i, used, matching, g);
    }

    vector<pair<int, int>> answ;
    for(int i =1; i<n+1;++i){
        if(matching[i]!=-1 and i<matching[i]) answ.push_back(pair(min(i, matching[i]), max(i, matching[i])));
    }
    sort(answ.begin(), answ.end());
    int answ_size = answ.size();
    cout<<answ_size;

    for(int i =0; i<answ_size;++i){
        cout<<'\n'<<answ[i].first<<' '<<answ[i].second;
    }
}
```

Функция обхода в глубину адаптированная под Алгоритм Куна. Тут можно обратить внимание, что в отличие от классической реализации мы добавляем в паросочетание обе вершины, т.е. восстанавливаем неорентированную связь. Чтобы не было повторений в ответе, при его формировании, мы убираем копии.

```
bool dfs(int idx, vector<int> &used, vector<int> &matching, vector<vector<int>> &graph){
    if (used[idx]!=0) return false;
    used[idx] = 1;
    for (int i = 0; i<graph[idx].size();++i){
        int to = graph[idx][i];
        if (matching[to] ==-1 or dfs(matching[to], used, matching, graph)) {
            matching[to] = idx;
            matching[idx] = to;
            return true;
        }

    }
    return false;

}
```

3 Консоль

```
arsenii@PC-Larcha14:~/Documents/C_pp_uk/DA/lab_9$ ./lab_9
4 3
1 2
2 3
3 4
2
1 2
3 4
```

4 Тест производительности

Сравним алгоритм Куна с решением задачи методом Форда-Фалкерсона.

```
arsenii@PC-Larcha14:~/Documents/C_pp_uk/DA/lab_9$ ./banchmark <test/tests_2.txt
n=100 m=100
Kuhn’s algorithm: 0.000207256 s
Fard-F algorithm: 0.001401426 s
arsenii@PC-Larcha14:~/Documents/C_pp_uk/DA/lab_9$ ./banchmark <test/tests_3.txt
n=100 m=10
Kuhn’s algorithm: 0.000161298 s
DP algorithm    : 0.000708145 s
```

Как можно заметить, алгоритм Куна работает быстрее на всех тестах, это связано с тем, что он был создан создан специально для решения задачи поиска максимального паросочетания, а метод Форда-Фалкерсона является адаптированным методом для решения данной задачи, т.к. он предназначен для поиска максимального потока в транспортной сети и следовательно выполняет лишние действия, которые тормозят работу алгоритма.

5 Выводы

Выполнив данную лабораторную работу, я освежил свои знания о графах, а кроме того, познакомился с задачей поиска максимального парочетания и смог реализовать алгоритм Куна для решения данной задачи.
Многие задачи связанные с графами очень часто встречаются в нашей жизни, поэтому знание алгоритмы, связанные с ними всегда полезны, не только для развития, но и просто в жизни.

Список литературы

[1] [1] Паросочетания: основные определения, теорема о максимальном паросочетании и дополняющих цепях [Электронный ресурс]: Википедия ITMO - URL https://neerc.ifmo.ru/wiki/inde