Московский авиационный институт
(Национальный исследовательский университет)
Институт №8
Кафедра 804

Курсовая работа

по курсу «Дискретный анализ»

Студент: Дюсекеев А. Е.
Группа: М8О-201М-21
Преподаватель: Рассказова В. А.
Оценка:

Москва, 2022

Задача: разработать алгоритм для нахождения множества путей в ориентированном ациклическом графе

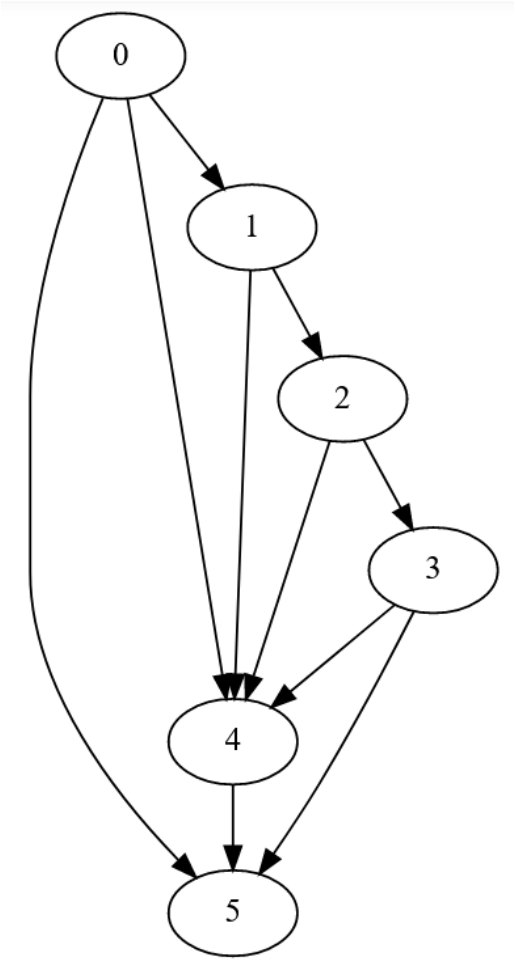
Код алгоритма:

```
import numpy as np
from graphviz import Digraph
class Graph:
    def __init__(self, adjacency):
        self.N = adjacency.shape[0]
        self.adjacency = adjacency
        self.v = list(range(self.N))
        self.graph = Digraph()
        self.paths = []
        for i in self.v:
            self.graph.node(f'{i}')
            for j in self.v:
                if adjacency[i, j] > 0:
                    self.graph.edge(f'{i}', f'{j}')

    def find_path(self, u, d, visited, path):
        visited[u]= True
        path.append(u)
        if u == d:
            self.paths.append(path.copy())
        else:
            for i in self.v:
                if self.adjacency[u, i] != 0 and not visited[i]:
                    self.find_path(i, d, visited, path)

        path.pop()
        visited[u] = False

    def find_all_paths(self, s, d):
        visited = [False]*(self.N)
        path = []
        self.find_path(s, d, visited, path)
        return self.paths

    def find_each_path(self):
        for i in self.v:
            for j in self.v:
                if i != j:
                    self.find_all_paths(i, j)
        return self.paths


a = np.array([
    [0, 1, 0, 0, 1, 1],
    [0, 0, 1, 0, 1, 0],
    [0, 0, 0, 1, 1, 0],
    [0, 0, 0, 0, 1, 1],
```

```
    [0, 0, 0, 0, 0, 1],
    [0, 0, 0, 0, 0, 0]
])
g = Graph(a)
g.find_each_path()
```

Результаты работы:

Исходный граф:



Множество всех путей:

```
[[0, 1],
 [0, 1, 2],
 [0, 1, 2, 3],
 [0, 1, 2, 3, 4],
 [0, 1, 2, 4],
 [0, 1, 4],
 [0, 4],
 [0, 1, 2, 3, 4, 5],
 [0, 1, 2, 3, 5],
 [0, 1, 2, 4, 5],
 [0, 1, 4, 5],
 [0, 4, 5],
 [0, 5],
 [1, 2],
 [1, 2, 3],
 [1, 2, 3, 4],
 [1, 2, 4],
 [1, 4],
 [1, 2, 3, 4, 5],
 [1, 2, 3, 5],
```

```
[1, 2, 4, 5],
[1, 4, 5],
[2, 3],
[2, 3, 4],
[2, 4],
[2, 3, 4, 5],
[2, 3, 5],
[2, 4, 5],
[3, 4],
[3, 4, 5],
[3, 5],
[4, 5]]
```