Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

Дисципліна: Алгоритми та методи обчислень

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2018

Лабораторная работа № 7.

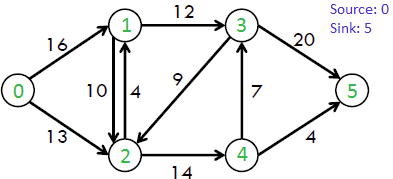
**Тема**: Задача про потоки. Алгоритм Форда-Фалкерсона.

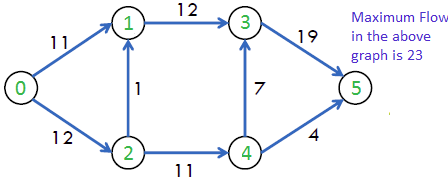
Задание:

Вариант №4

Найти максимальный поток груза, который можно пропустить через заданную

транспортную сеть.





import collections

‘’’ Этот класс представляет собой ориентированный граф с использованием представления матрицы смежности’’’

class Graph:

def \_\_init\_\_(self,graph):

self.graph = graph # residual graph

self.ROW = len(graph)

def BFS(self, s, t, parent):

visited = [False] \* (self.ROW)

queue = collections.deque()

queue.append(s)

visited[s] = True

while queue:

u = queue.popleft()

for ind, val in enumerate(self.graph[u]):

if (visited[ind] == False) and (val > 0):

queue.append(ind)

visited[ind] = True

parent[ind] = u

return visited[t]

def FordFulkerson(self, source, sink):

parent = [-1] \* (self.ROW)

max\_flow = 0

while self.BFS(source, sink, parent):

path\_flow = float("Inf")

s = sink

while s != source:

path\_flow = min(path\_flow, self.graph[parent[s]][s])

s = parent[s]

max\_flow += path\_flow

v = sink

while v != source:

u = parent[v]

self.graph[u][v] -= path\_flow

self.graph[v][u] += path\_flow

v = parent[v]

return max\_flow

if \_\_name\_\_ == "\_\_main\_\_":

adj\_matrix = [[0, 16, 13, 0, 0, 0],

[0, 0, 10, 12, 0, 0],

[0, 4, 0, 0, 14,0],

[0, 0, 9, 0, 0, 20],

[0, 0, 0, 7, 0, 4],

[0, 0, 0, 0, 0, 0]]

source = 0

sink = 5

graph = Graph(adj\_matrix)

max\_flow = graph.FordFulkerson(source, sink)

print("Максимальная пропускная способность %d " % max\_flow)

