Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 4

по дисциплине «УИР»

Выполнил: студент группу ИСТ-22-2б Поляков П.Е.

Проверил: ассистент каф. ВММБ Богданова А.П.

Пермь, 2023

Содержание

[Задание 1 3](#_Toc151926861)

[1.1 . Постановка задачи 3](#_Toc151926862)

[1.2. Решение задачи, код программы 3](#_Toc151926863)

[1.3. Тестирование работы программы с проверкой 4](#_Toc151926864)

[Задание 2 5](#_Toc151926865)

[2.1. Постановка задачи 5](#_Toc151926866)

[2.2. Решение задачи, код программы 5](#_Toc151926867)

[2.3. Тестирование работы программы с проверкой 6](#_Toc151926868)

[Задание 3 7](#_Toc151926869)

[3.1. Постановка задачи 7](#_Toc151926870)

[3.2. Решение задачи, код программы 7](#_Toc151926871)

[Задание 4 8](#_Toc151926872)

[4.1. Постановка задачи 8](#_Toc151926873)

[4.2. Решение задачи, код программы 8](#_Toc151926874)

[Задание 5 9](#_Toc151926875)

[5.1. Постановка задачи 9](#_Toc151926876)

[5.2. Решение задачи, код программы 9](#_Toc151926877)

[Задание 6 10](#_Toc151926878)

[6.1. Постановка задачи 10](#_Toc151926879)

[6.2. Решение задачи, код программы 10](#_Toc151926880)

[6.3. Тестирование работы программы с проверкой 12](#_Toc151926881)

# Задание 1

## . Постановка задачи

Найдите длину кратчайшего пути от начального до конечного узла.

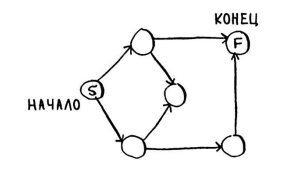


Рисунок №1. Граф

## 1.2. Решение задачи, код программы

graph = {  
 'A': {'B': 7, 'C': 4},  
 'B': {'A': 7, 'D': 10, 'F': 7},  
 'C': {'A': 4, 'D': 6, 'G': 3},  
 'D': {'B': 10, 'C': 6},  
 'G': {'C': 3, 'F': 8},  
 'F': {'B': 7, 'G': 8}  
}  
  
  
def shortestPath(start, end):  
 global graph  
 shortest\_paths = {node: float('inf') for node in graph}  
 shortest\_paths[start] = 0  
 unvisited\_nodes = set(graph.keys())  
  
 while unvisited\_nodes:  
 current\_node = min(unvisited\_nodes, key=lambda node: shortest\_paths[node])  
 unvisited\_nodes.remove(current\_node)  
  
 for neighbor, weight in graph[current\_node].items():  
 if shortest\_paths[neighbor] > shortest\_paths[current\_node] + weight:  
 shortest\_paths[neighbor] = shortest\_paths[current\_node] + weight  
  
 return shortest\_paths[end]  
  
  
def main():  
 print("Минимальная длина от А до F:\t", shortestPath('A','F'))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

## 1.3. Тестирование работы программы с проверкой

Для проверки задачи были произведены запуска программы, их результаты можно увидеть в таблице №1

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Python |
| 1 |  |

Данным снимком экрана, я показал что программа работает верно.

# Задание 2

## 2.1. Постановка задачи

Найдите длину кратчайшего пути от «саb» к «bat».

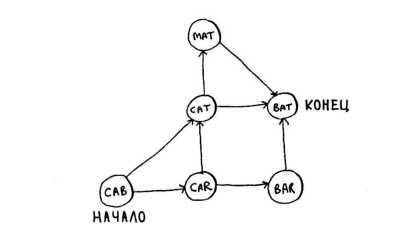


Рисунок №2. Граф

## 2.2. Решение задачи, код программы

from collections import deque

def bfs(graph, start, end):

queue = deque([(start, [start])])

visited = set()

while queue:

current, path = queue.popleft()

if current == end:

return path

if current not in visited:

visited.add(current)

for neighbor in graph[current]:

if neighbor not in visited:

queue.append((neighbor, path + [neighbor]))

return None # Если путь не найден

# Пример использования

graph = {

'CAB': ['CAR', 'CAT'],

'CAR': ['CAT', 'BAR'],

'CAT': ['MAT', 'BAT'],

'BAR': ['MAT'],

'MAT': ['BAT']

}

start\_node = 'CAB'

end\_node = 'BAT'

shortest\_path = bfs(graph, start\_node, end\_node)

if shortest\_path:

print(f"Кратчайший путь от {start\_node} до {end\_node}: {shortest\_path}")

else:

print(f"Путь от {start\_node} до {end\_node} не найден.")

## 2.3. Тестирование работы программы с проверкой

Для проверки задачи были произведены запуски программы, их результаты можно увидеть в таблице №2

Таблица 2

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Python |
| 1 |  |
| № п.п. | Решение вручную |
| 1 |  |

Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 3

## 3.1. Постановка задачи

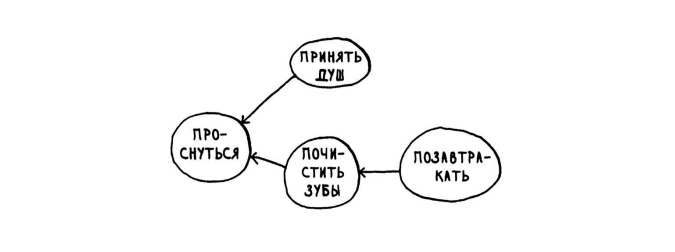


Рисунок №3. Граф

Для каждого из следующих трех списков укажите, действителен он или недействителен.

|  |  |  |
| --- | --- | --- |
| А | Б | В |
| 1. Проснуться 2. Принять душ 3. Позавтракать 4. Почистить зубы | 1. Проснуться 2. Почистить зубы 3. Позавтракать 4. Принять душ | 1. Принять душ 2. Проснуться 3. Почистить зубы 4. Позавтракать |

## 3.2. Решение задачи, код программы

Списки А и Б действительны, список В не является действительным в соответствии с графом (рисунок №1)

# Задание 4

## 4.1. Постановка задачи

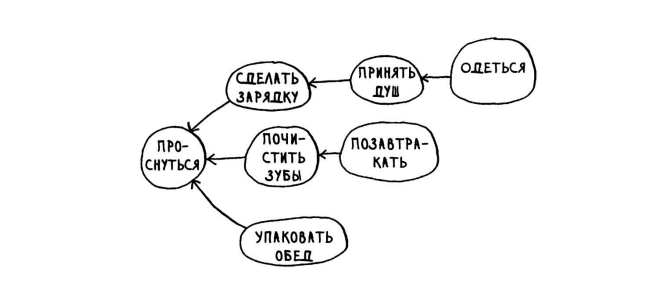


Рисунок №4. Увеличенный граф

Постройте действительный список для этого графа.

## 4.2. Решение задачи, код программы

|  |  |  |
| --- | --- | --- |
| А | Б | В |
| 1. Проснуться 2. Сделать зарядку 3. Принять душ 4. Одеться 5. Почистить зубы 6. Позавтракать 7. Упаковать обед | 1. Проснуться 2. Сделать зарядку 3. Принять душ 4. Одеться 5. Упаковать обед 6. Почистить зубы 7. Позавтракать | 1. Проснуться 2. Почистить зубы 3. Позавтракать 4. Сделать зарядку 5. Принять душ 6. Одеться 7. Упаковать обед |
| Г | Д | Е |
| 1. Проснуться 2. Почистить зубы 3. Позавтракать 4. Упаковать обед 5. Сделать зарядку 6. Принять душ 7. Одеться | 1. Проснуться 2. Упаковать обед 3. Сделать зарядку 4. Принять душ 5. Одеться 6. Почистить зубы 7. Позавтракать | 1. Проснуться 2. Упаковать обед 3. Почистить зубы 4. Позавтракать 5. Сделать зарядку 6. Принять душ 7. Одеться |

# Задание 5

## 5.1. Постановка задачи

Какие из следующих графов также являются деревьями?

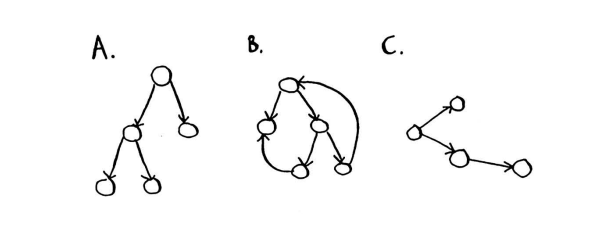


Рисунок №5. Деревья

## 5.2. Решение задачи, код программы

Деревом называют особую разновидность графа, в котором нет ребер, указывающих в обратном направлении. В соответствии с этим определением деревьями являются А и Б.

# Задание 6

## 6.1. Постановка задачи

Заполнить массив и создать его зеркала в соответствии с рисунком №4

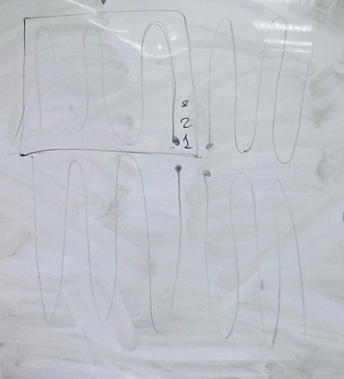


Рисунок №4. Схема

## 6.2. Решение задачи, код программы

matrix = []

def generate\_even\_odd\_matrix(rows, cols):

global matrix

matrix = [[0] \* cols for \_ in range(rows)]

count = 1

for i in range(cols - 1, -1, -1):

if i % 2 == 0:

for j in range(rows - 1, -1, -1):

matrix[j][i] = count

count += 1

else:

for j in range(rows):

matrix[j][i] = count

count += 1

return matrix

def generate\_odd\_even\_matrix(rows, cols):

global matrix

matrix = [[0] \* cols for \_ in range(rows)]

count = 1

for i in range(cols - 1, -1, -1):

if i % 2 == 1:

for j in range(rows - 1, -1, -1):

matrix[j][i] = count

count += 1

else:

for j in range(rows):

matrix[j][i] = count

count += 1

return matrix

def mirror\_bottom\_left():

array\_bottom\_left = matrix[::-1]

return array\_bottom\_left

def mirror\_bottom\_right(array\_bottom\_left):

array\_bottom\_right = [row[::-1] for row in array\_bottom\_left]

return array\_bottom\_right

def mirror\_upper\_right(array\_bottom\_right):

array\_upper\_right = array\_bottom\_right[::-1]

return array\_upper\_right

def build\_matrix(rows, cols):

if cols % 2 == 0:

generate\_odd\_even\_matrix(rows, cols)

else:

generate\_even\_odd\_matrix(rows, cols)

def main():

rows = int(input("Введите количество строк матрицы:\n"))

while rows < 1:

print("Вы допустили ошибку, повторите ввод")

rows = int(input())

cols = int(input("Введите количество столбцов матрицы:\n"))

while cols < 1:

print("Вы допустили ошибку, повторите ввод")

cols = int(input())

print("Основная матрица:")

build\_matrix(rows, cols)

for i in matrix:

print(' '.join(map(str, i)))

print("Преобразование - Слева-снизу")

for i in mirror\_bottom\_left():

print(' '.join(map(str, i)))

print("Преобразование - Справа-снизу")

for i in mirror\_bottom\_right(mirror\_bottom\_left()):

print(' '.join(map(str, i)))

print("Преобразование - Справа-сверху")

for i in mirror\_upper\_right(mirror\_bottom\_right(mirror\_bottom\_left())):

print(' '.join(map(str, i)))

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

main()

## 6.3. Тестирование работы программы с проверкой

Для проверки задачи были произведены запуски программы, их результаты можно увидеть в таблице №3

Таблица 3

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Python |
| 1 |  |

Данным снимком показал, что программа работает верно.