**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине: **«Оптимизация проектных решений»**

на тему: **«Дискретное программирование. Задача о коммивояжере»**

Выполнил: студент гр. ИТП-41

Морозов Д.А.

Принял: преподаватель

Мурашко И. А.

Гомель 2020

**Цель**: знакомство с методами решения задач дискретного программирования.

**Ход работы**

1. Изучить методы решения задач дискретного программирования – метод ветвей и границ и метод ближайшего соседа.

2. Разработать программы, решающие задачу о коммивояжере методом ветвей и границ и методом ближайшего соседа.

3. Решить задачу из задания 1 на бумаге сравнить полученные результаты.

4. Решить задачи из задания 1 и 2 при помощи разработанных программ.

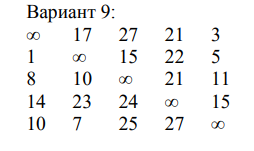


Рисунок 1 – Индивидуальное задание

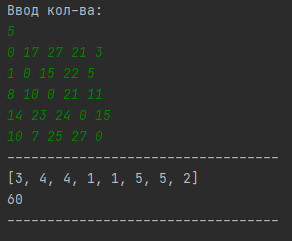


Рисунок 2 – Результат нахождения значения целевой функции

**Вывод:** в ходе выполнения лабораторной работы 3 были изучены и разработаны алгоритмы для решения задачи о коммивояжере. Освоены основные навыки применения алгоритмов для данного типа задач.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

**Algorithm.py**

#Функция нахождения минимального элемента, исключая текущий элемент

def Min(lst,myindex):

return min(x for idx, x in enumerate(lst) if idx != myindex)

#функция удаления нужной строки и столбцах

def Delete(matrix,index1,index2):

del matrix[index1]

for i in matrix:

del i[index2]

return matrix

#Функция вывода матрицы

def PrintMatrix(matrix):

print("---------------")

for i in range(len(matrix)):

print(matrix[i])

print("---------------")

print("Ввод кол-ва: ")

n=int(input())

matrix=[]

H=0

PathLenght=0

Str=[]

Stb=[]

res=[]

result=[]

StartMatrix=[]

#Инициализируем массивы для сохранения индексов

for i in range(n):

Str.append(i)

Stb.append(i)

#Вводим матрицу

for i in range(n): matrix.append(list(map(int, input().split())))

#Сохраняем изначальную матрицу

for i in range(n):StartMatrix.append(matrix[i].copy())

#Присваеваем главной диагонали float(inf)

for i in range(n): matrix[i][i]=float('inf')

while True:

#Редуцируем

#--------------------------------------

#Вычитаем минимальный элемент в строках

for i in range(len(matrix)):

temp=min(matrix[i])

H+=temp

for j in range(len(matrix)):

matrix[i][j]-=temp

#Вычитаем минимальный элемент в столбцах

for i in range(len(matrix)):

temp = min(row[i] for row in matrix)

H+=temp

for j in range(len(matrix)):

matrix[j][i]-=temp

#--------------------------------------

#Оцениваем нулевые клетки и ищем нулевую клетку с максимальной оценкой

#--------------------------------------

NullMax=0

index1=0

index2=0

tmp=0

for i in range(len(matrix)):

for j in range(len(matrix)):

if matrix[i][j]==0:

tmp=Min(matrix[i],j)+Min((row[j] for row in matrix),i)

if tmp>=NullMax:

NullMax=tmp

index1=i

index2=j

#--------------------------------------

#Находим нужный нам путь, записываем его в res и удаляем все ненужное

res.append(Str[index1]+1)

res.append(Stb[index2]+1)

oldIndex1=Str[index1]

oldIndex2=Stb[index2]

if oldIndex2 in Str and oldIndex1 in Stb:

NewIndex1=Str.index(oldIndex2)

NewIndex2=Stb.index(oldIndex1)

matrix[NewIndex1][NewIndex2]=float('inf')

del Str[index1]

del Stb[index2]

matrix=Delete(matrix,index1,index2)

if len(matrix)==1:break

#Формируем порядок пути

for i in range(0,len(res)-1,2):

if res.count(res[i])<2:

result.append(res[i])

result.append(res[i+1])

for i in range(0,len(res)-1,2):

for j in range(0,len(res)-1,2):

if result[len(result)-1]==res[j]:

result.append(res[j])

result.append(res[j+1])

print("----------------------------------")

print(result)

#Считаем длину пути

for i in range(0,len(result)-1,2):

if i==len(result)-2:

PathLenght+=StartMatrix[result[i]-1][result[i+1]-1]

PathLenght+=StartMatrix[result[i+1]-1][result[0]-1]

else: PathLenght+=StartMatrix[result[i]-1][result[i+1]-1]

print(PathLenght)

print("----------------------------------")

input()