ГУАП

КАФЕДРА № 52

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | С А Митрофанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О КУРСОВОЙ РАБОТЕ |
| Задача о назначениях |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 5721 |  |  |  | А. Р. Осиков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

Оглавление

[Постановка задачи 3](#_Toc532815923)

[Алгоритм 3](#_Toc532815924)

[Пример 3](#_Toc532815925)

[Псевдокод 6](#_Toc532815926)

[Сложность 6](#_Toc532815927)

[Инструкция пользователя 7](#_Toc532815928)

[Параметры запускаемой строки 7](#_Toc532815929)

[Формат входного и выходного файлов 7](#_Toc532815930)

[Тестовые примеры 7](#_Toc532815931)

[Пример для консоли: 7](#_Toc532815932)

[Пример №1 7](#_Toc532815933)

[Пример №2 8](#_Toc532815934)

[Пример №3 9](#_Toc532815935)

[Список литературы 11](#_Toc532815936)

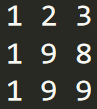
# 

# Постановка задачи

Задачей данной курсовой работы является разработка программы, которая способна решать задачу о назначениях двумя способами: жадным алгоритмом и с помощью Венгерского алгоритма.

На вход обоих алгоритмов подается квадратная матрица работник - работа. В каждой i-ой строке этой матрицы указаны варианты оплаты за j-ую работу для i-ого работника. Алгоритмам необходимо поставить каждого работника на одну из работ так чтобы каждой работе соответствовал один работник. При этом им необходимо найти самый дешевый вариант из всех комбинаций.

Например, дана матрица 3х3:



В данном случае можно просто идти постепенно от работника к работнику и выбирать для них самую выгодную незанятую работу: для 1 – 1, для 2 – 3, и для 3 – 2. В результате мы получим общую стоимость: 18. А теперь проверим, может это не самый лучший вариант, и действительно вариант 1 – 2, 2 – 3, 3 – 1 даст нам более выгодный вариант, вместо 17 мы получим всего 11.

# Алгоритм

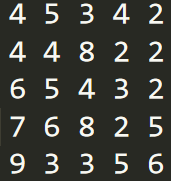
Алгоритм жадного перебора построен на идее того, что мы хотим максимально быстро получить неплохой вариант решения задачи, на каждом шаге ставя одному работнику самую дешевую незанятую работу, скорее всего он не будет оптимальным для данной матрицы.

Венгерский алгоритм при помощи упрощения матрицы находит самый оптимальный вариант для данной матрицы, но при этом работает значительно дольше чем жадный перебор.

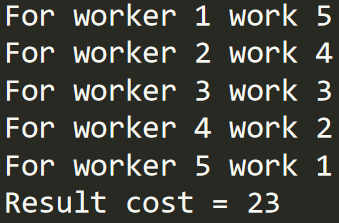
Рассмотрим работу алгоритмов на следующем примере:

## Пример

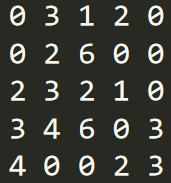
Возьмем матрицу 5х5:



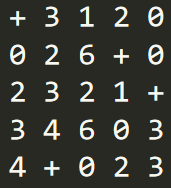
На первом шаге жадный алгоритм выберет в первой строке пятый столбец, потому что там самый дешевый вариант для первого работника, далее для второй строки будет выбран четвертый столбец, и т.д. В результате работы алгоритма мы получим:



Теперь рассмотрим, как будет действовать Венгерский алгоритм. Этот алгоритм можно разделить на 2 шага, повторяя которые мы когда-нибудь получим оптимальный вариант. Для начала рассмотрим первый шаг, на нем следует вычесть из всех строк минимальное значение в текущей строке, далее вычесть из всех столбцов минимальное значение в текущем столбце. Если у нас появилась хотя бы одна комбинация из нулей, то это и есть оптимальный вариант, это можно проверить при помощи алгоритма нахождения максимального количества паросочетания в двудольном графе, в данном случае используется алгоритм Куна. Во многих случаях мы достигаем желаемого результата уже после первого шага, но иногда это не так, в нашем примере после первого шага получится матрица:

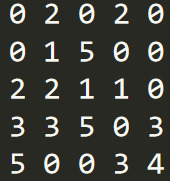


Алгоритм Куна в данном случае даст только 4 пары, чего недостаточно, после шага 1 следует шаг 2. Он заключается в том, что сначала назначается как можно больше работников на нулевую стоимость



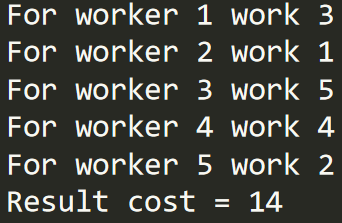
Далее отмечаем все строки, в которых нет помеченных нулей (отмечаем 4 строку), далее выбираем все столбцы с нулями в этих строках (отмечаем 4 столбец), далее отмечаем все строки с помеченными нулями (отмечаем 2 строку), продолжаем, пока новые строки и столбцы не перестанут появляться, помечаем 1 и 5 столбцы и наконец помечаем 1 и 3 строки. Теперь рассмотрим пересечение неотмеченных столбцов и отмеченных строк, найдем из этих чисел самое минимальное, после этого уменьшим все эти числа на это число, а числа, которые находятся на пересечении отмеченных столбцов и неотмеченных строк увеличим на то самое минимальное число.

После второго шага следует проверить не нашлась ли комбинация из нулей, в нашем случае хватило одного повторения двух шагов.



Иногда шаг 2 не дает результата, тогда следует повторить шаг 1, и, если снова нет результата повторять шаги поочередно пока не появится комбинация из нулей, она и будет оптимальным решением.

В нашем случае мы получили такое решение:



Для этого решения было затрачено больше усилий и времени чем для решения полученного с помощью жадного алгоритма, но в результате мы получили оптимальное решение, и оно меньше, чем результат жадного алгоритма.

## Псевдокод

Жадный алгоритм:

Цикл i по длине матрицы

Цикл j по ширине матрицы

Нахождение оптимальной j-ой работы для i-ого работника

Венгерский алгоритм:

Цикл пока не найдено решение с помощью алгоритма Куна

Шаг 1

Проверка не найдено ли решения с помощью алгоритма Куна

Шаг 2

## Сложность

На основе псевдокода возможно оценить сложность.

В случае жадного алгоритма имеется 2 цикла и оба зависят от размерности матрицы, из этого следует, что сложность жадного алгоритма равна O(n^2)

В случае Венгерского алгоритма сложность зависит от сложности алгоритма Куна, в свою очередь его сложность зависит от количества вершин и количества ребер и выражается в виде O(m \* n), что в худшем случае O(n^3), Алгоритм куна будет вызван не больше 2n раз, в следствии чего, Венгерскому алгоритму можно дать оценку O(n^4)

# Инструкция пользователя

## Параметры запускаемой строки

Для работы с программой пользователю необходимо запустить исполняемый файл и далее ввести имя исходного файла и 2-ух результирующих.

## Формат входного и выходного файлов

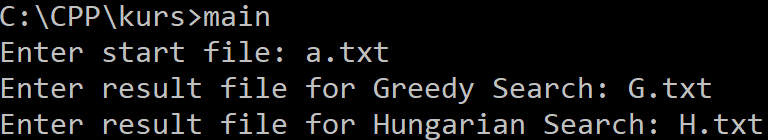
Входной файл имеет следующий вид:

1) На первой строке располагается число соответствующие размеру матрицы.

2) На остальных строках располагается матрица, одной строке соответствует одна строка матрицы, числа следует вводить через пробел.

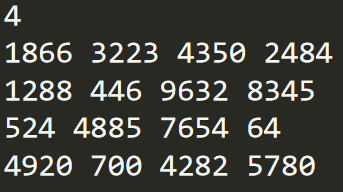
# Тестовые примеры

### Пример для консоли:



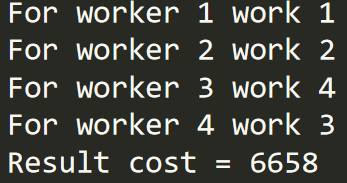
### Пример №1

Входной файл:

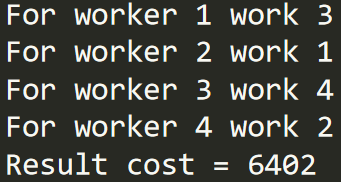


Выходные файлы:

G.txt:

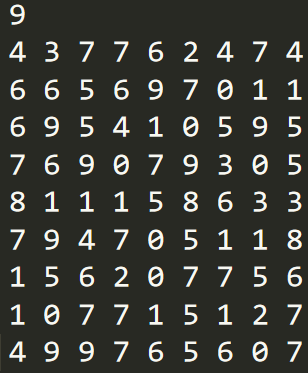


H.txt:



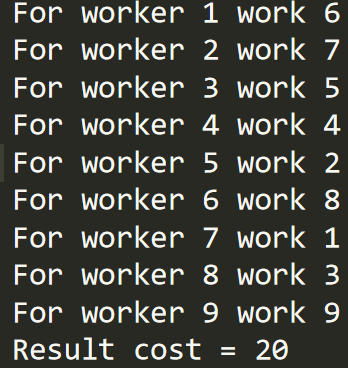
### Пример №2

Входной файл:

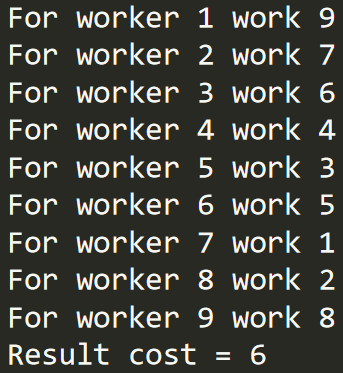


Выходные файлы:

G.txt:

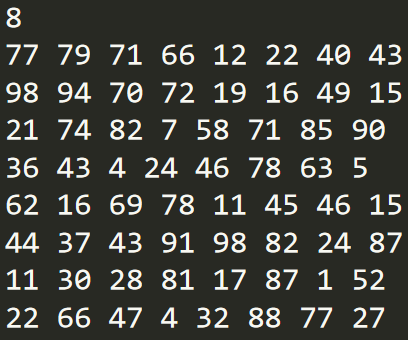


H.txt:



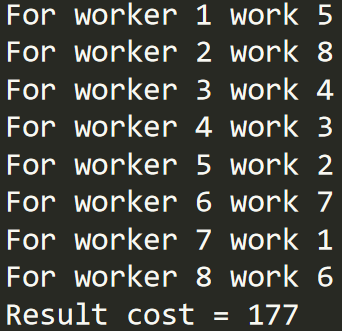
### Пример №3

Входной файл:

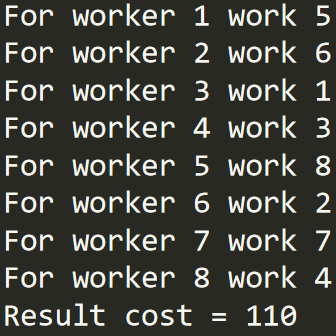


Выходные файлы:

G.txt:



H.txt:



# Список литературы

Книги:

Э. Рейнгольд, Ю. Нивергельт, Н. Део, Комбинаторные алгоритмы, Мир, 1980г.

А. Кофман, ВВедение в прикладную комбинаторику, Наука, 1975г.

Сайты:

[Задача о назначениях - Википедия](https://ru.wikipedia.org/wiki/Задача_о_назначениях)

[Венгерский алгоритм - Википедия](https://ru.wikipedia.org/wiki/Венгерский_алгоритм)

[Алгоритм Куна - MAXimal](http://e-maxx.ru/algo/kuhn_matching)