Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт расчетной работе № 1**

Дисциплина: Системный анализ и принятие решений.

Выполнил студент гр. 5130901/10101 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Л. Симоновский (подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Г. Сиднев (подпись)

“19” февраля 2023 г.

Санкт-Петербург

2024

Оглавление

[1. Условие: 2](#_Toc159275011)

[1.1. Вариант: 2](#_Toc159275012)

[1.2. Условие задания: 2](#_Toc159275013)

[2. Ход решения 3](#_Toc159275014)

[2.1. Определить наиболее ранние моменты начала работ с использованием метода математического программирования: 3](#_Toc159275015)

[2.2. Определить наиболее ранние моменты начала работ и их интенсивности, если длительность равна интенсивности выполнения работ, а суммарная интенсивность не превышает 75% от общего числа выполняемых работ. 4](#_Toc159275016)

[2.3. Самостоятельно распределить работы между заданным числом исполнителей и сформулировать задачу математического программирования с бинарными индикаторными переменными. Определить число бинарных переменных и дополнительных ограничений в этой задаче и дать содержательную формулировку части ограничений с бинарными переменными. 6](#_Toc159275017)

[2.3.1. Изменить формулировку задачи так, чтобы число бинарных переменных не превышало 10. Решить полученную задачу с использованием команды intlinprog. Определить мощность множества бинарных переменных задачи и дать содержательную интерпретацию полученному решению. 7](#_Toc159275018)

[2.4. Найти характеристики , и расписания выполнения комплекса работ с использованием метода динамического программирования. Привести соответствующие уравнения Беллмана. Определить критические пути на графе. 10](#_Toc159275019)

[2.5. Найти те же характеристики , и расписания выполнения комплекса работ с использованием математического программирования. 11](#_Toc159275020)

[2.6. Определить помимо полных резервов времени работ резервы времени, относящиеся к событиям сетевого графа, а именно . 13](#_Toc159275021)

[3. Вывод 14](#_Toc159275022)

[4. Ссылки 14](#_Toc159275023)

[5. Приложение 14](#_Toc159275024)

# Условие:

## Вариант:

Вариант №10.

Граф №19.

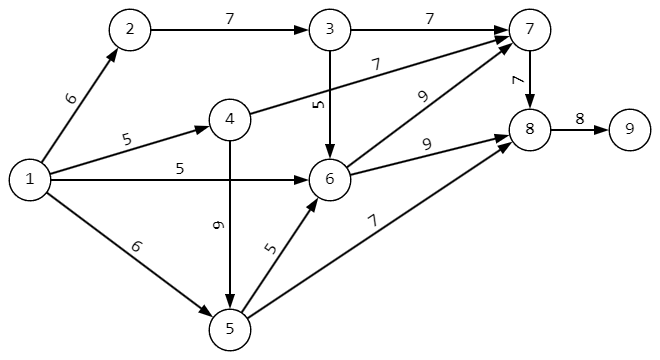


Рис. 1.1. Граф №19.

Число исполнителей 2.

Решающее правило: Короткие работы вперед.

## Условие задания:

Выполнить следующие разделы:

1. Определить наиболее ранние моменты начала работ с использованием метода математического программирования.
2. Определить наиболее ранние моменты начала работ и их интенсивности, если длительность равна интенсивности выполнения работ, а суммарная интенсивность не превышает 75% от общего числа выполняемых работ.
3. Самостоятельно распределить работы между заданным числом исполнителей и сформулировать задачу математического программирования с бинарными индикаторными переменными. Определить число бинарных переменных и дополнительных ограничений в этой задаче и дать содержательную формулировку части ограничений с бинарными переменными.
   1. Изменить формулировку задачи так, чтобы число бинарных переменных не превышало 10. Решить полученную задачу с использованием команды **intlinprog**. Определить мощность множества бинарных переменных задачи и дать содержательную интерпретацию полученному решению.
4. Найти характеристики , и расписания выполнения комплекса работ с использованием метода динамического программирования. Привести соответствующие уравнения Беллмана. Определить критические пути на графе.
5. Найти те же характеристики , и расписания выполнения комплекса работ с использованием математического программирования.
6. Определить помимо полных резервов времени работ резервы времени, относящиеся к событиям сетевого графа, а именно *.*
7. Рассмотреть вероятностную постановку задачи анализа расписания.

Считать СКО времен выполнения работ равными 5% от их длительностей. Предполагая неизменным критический путь (оценить справедливость этого предположения) найти вероятность того, что время выполнения комплекса работ не превысит найденного для детерминированной задачи в п.1 на 10%.

1. Представить пошаговую процедуру имитационного моделирования расписания по схеме событий с учетом числа исполнителей и решающего правила ранжирования работ из числа возможных. По результатам моделирования построить диаграмму Гантта.

# Ход решения

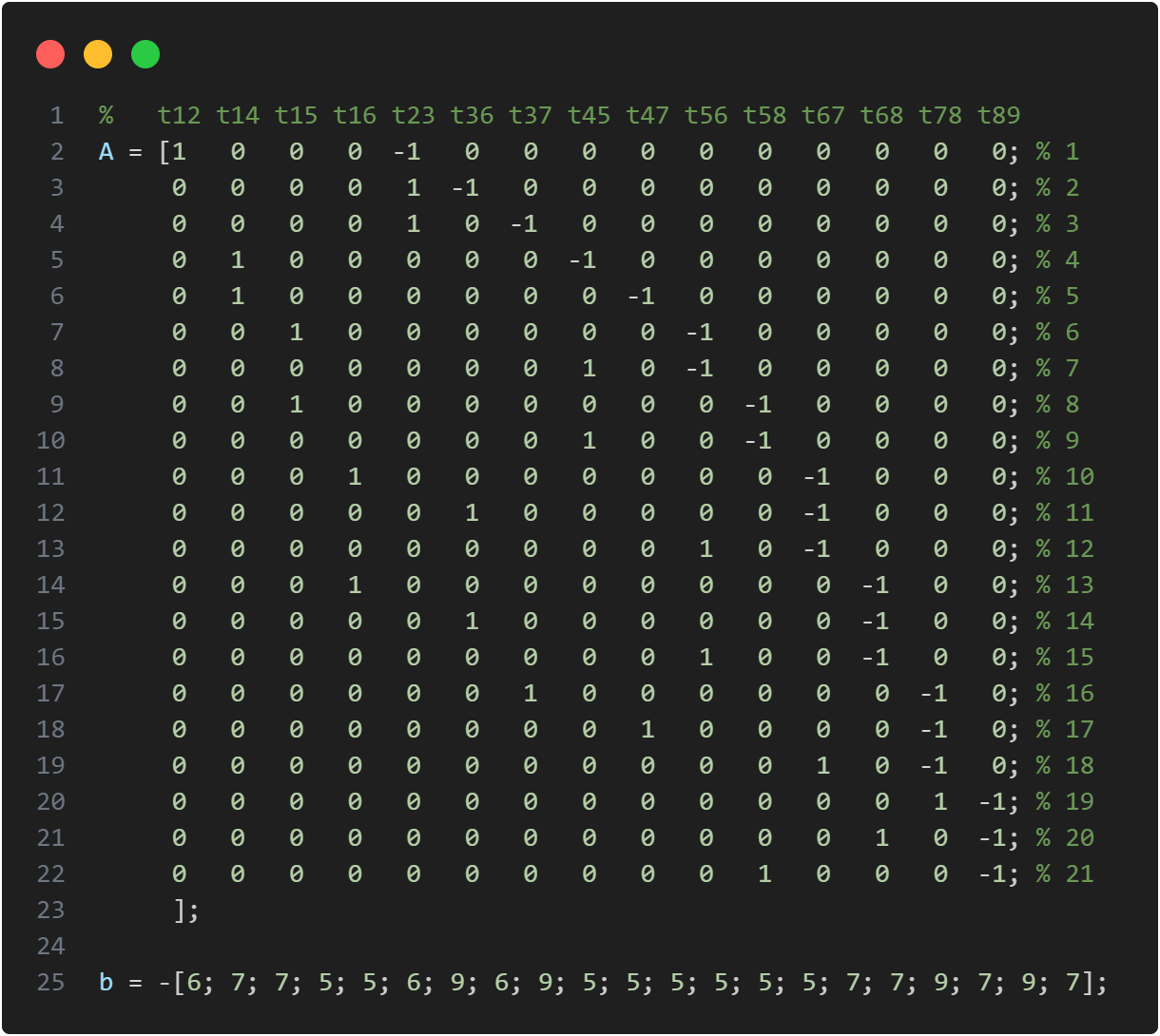
## Определить наиболее ранние моменты начала работ с использованием метода математического программирования:

Для графа, представленного на Рис. 1.1, составим систему неравенств для последующего решения с помощью методов линейного программирования. Обозначим за наиболее ранний момент начала работы , а за – наиболее ранний момент окончания всех работ.

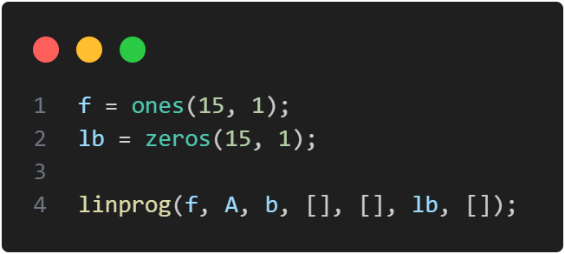
|  |  |  |
| --- | --- | --- |
|  |  |  |

Задача оптимизации – минимизация следующей функции:

Решим эту задачу с помощью функции Matlab linprog. Для этого преобразуем полученные ранее ограничения в матрицы и :



После чего вызовем linprog:



Полученный результат выглядит следующим образом:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 6 | 13 | 13 | 5 | 5 | 14 | 14 | 19 | 19 | 28 | 35 |

Табл. 2.1. Время начала всех работ.

Теперь мы знаем минимальное время начала каждой работы. Для получения информации о времени выполнения всех работ необходимо к времени начала работы прибавить время её выполнения т.е. 8.

Итого время выполнения всех работ равно 43.

## Определить наиболее ранние моменты начала работ и их интенсивности, если длительность равна интенсивности выполнения работ, а суммарная интенсивность не превышает 75% от общего числа выполняемых работ.

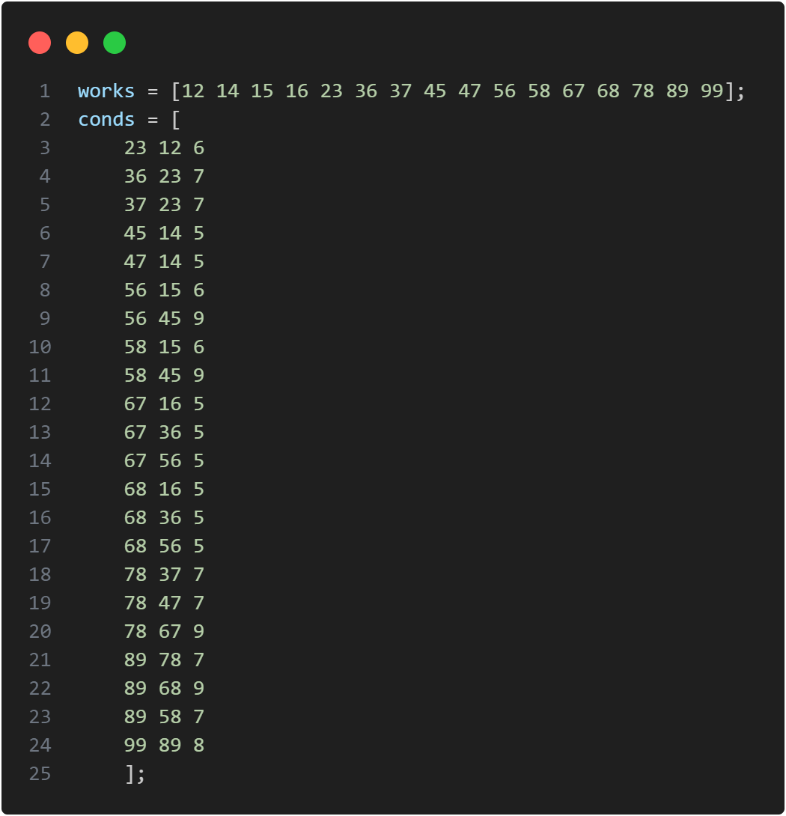
Мы сможем увеличить время выполнения всех работ за счет добавления интенсивностей работ, отличных от 1 – некоторые работы ускорим (интенсивность > 1), а некоторые замедлим (интенсивность < 1), если это потребуется.

Изменим исходную систему неравенств согласно правилу:

где – интенсивность работы.

Такими условиями мы пытаемся минимизировать время начала всех работ, при интенсивности, не превышающей 75% от числа выполняемых работ т.е. 15.

Создадим набор всех «работ» т.е. ребер графа и массив троек, где закодируем систему неравенств, созданную ранее:



Стоит заметить, что появилась работа-фальшивка. Это необходимо, чтоб MATLAB оптимизировал также и путь из 8 в 9 вершину и выводил нам результат этой оптимизации.

Создадим необходимые параметры для fmincon, а также функцию, которая распарсит заданные нам тройки в требуемые для fmincon значения и выведем результат выполнения на экран:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

В результате выполнения получим следующие значения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Моменты начала работ | |  | Интенсивности | |
|  | 0.00000 |  |  | 1.3138 |
|  | 0.00000 |  |  | 1.2574 |
|  | 0.00000 |  |  | 0.5536 |
|  | 0.00000 |  |  | 0.2890 |
|  | 4.56700 |  |  | 1.2735 |
|  | 10.0636 |  |  | 0.6906 |
|  | 10.0636 |  |  | 0.4130 |
|  | 3.97630 |  |  | 1.3114 |
|  | 3.97630 |  |  | 0.3039 |
|  | 10.8390 |  |  | 0.7734 |
|  | 10.8390 |  |  | 0.2723 |
|  | 17.3038 |  |  | 0.9269 |
|  | 17.3038 |  |  | 0.4677 |
|  | 27.0136 |  |  | 0.7344 |
|  | 36.5455 |  |  | 0.6692 |
|  | 48.5008 |  |  |  |

Табл. 2.2. Результат выполнения программы.

Сумма интенсивностей равна 11.25, что составляет ровно 75% от числа исполняемых работ, как и требовалось в задании. Стоит отметить, что общее время работы возрастало до 48.5 с 43 т.е. на 5.5 секунды (или в 1,13 раза). Это связано с тем, что при уменьшении интенсивности, некоторые работы стали работать дольше.

## Самостоятельно распределить работы между заданным числом исполнителей и сформулировать задачу математического программирования с бинарными индикаторными переменными. Определить число бинарных переменных и дополнительных ограничений в этой задаче и дать содержательную формулировку части ограничений с бинарными переменными.

Распределим 15 работ по двум исполнителям следующим образом:

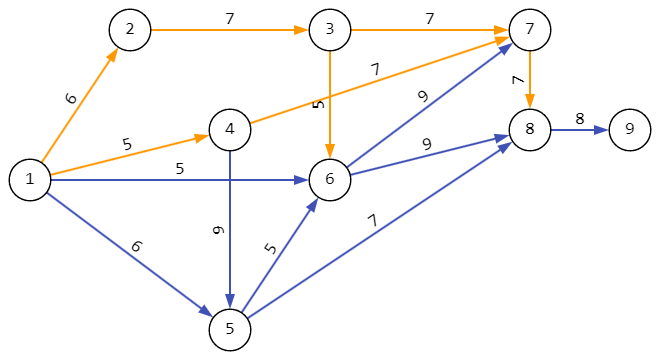


Рис. 2.1. Граф с задачами, распределенными по исполнителям.

Таким образом желтый цвет – первый исполнитель (7 задач), а синий – второй (8 задач).

Составим следующую систему для каждой пары работ

где , тогда число дополнительных ограничений задачи c бинарными переменными будет равно , а число бинарных переменных .

### Изменить формулировку задачи так, чтобы число бинарных переменных не превышало 10. Решить полученную задачу с использованием команды intlinprog. Определить мощность множества бинарных переменных задачи и дать содержательную интерпретацию полученному решению.

Упростим поставленную задачу, пусть только некоторые задачи выполняются определенным исполнителем, а над остальными может работать неограниченное число исполнителей:

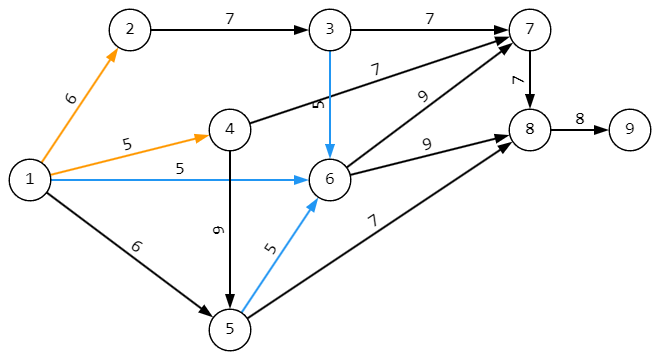
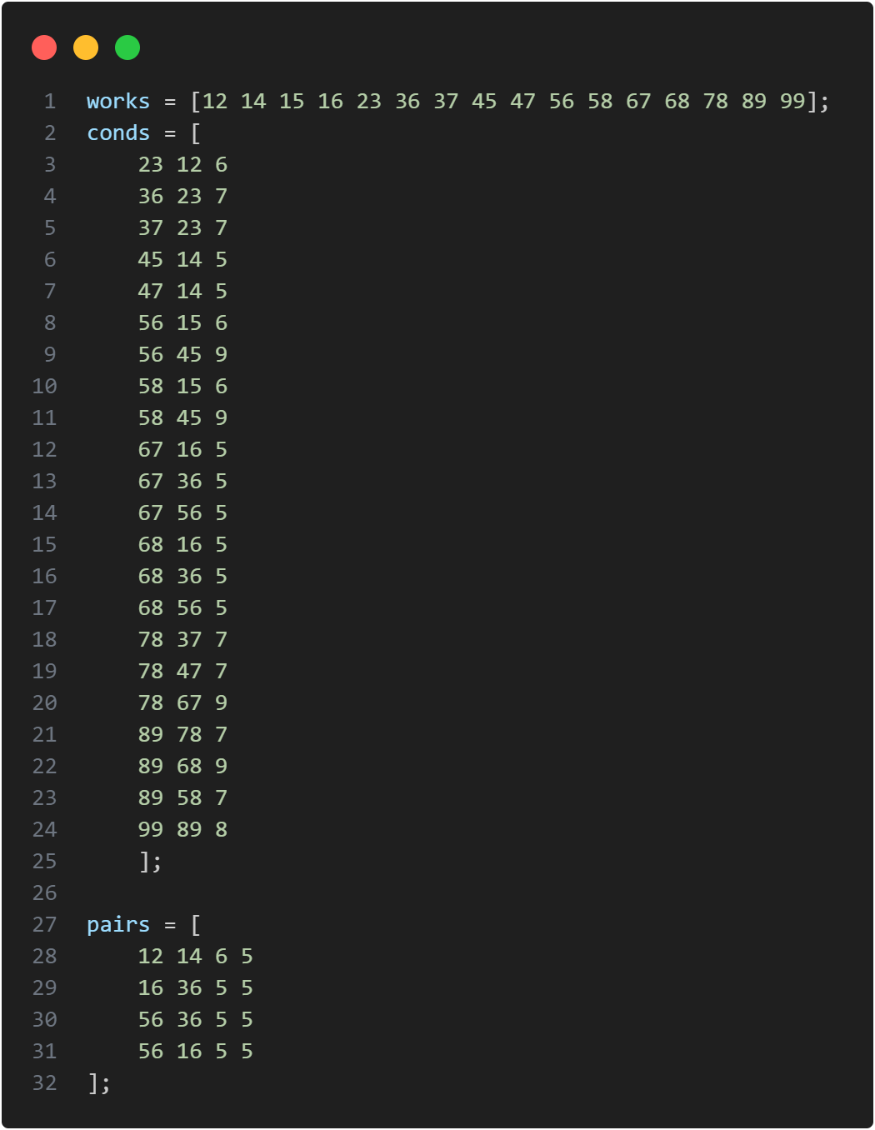


Рис. 2.2. Граф с задачами, часть которых распределена по исполнителям.

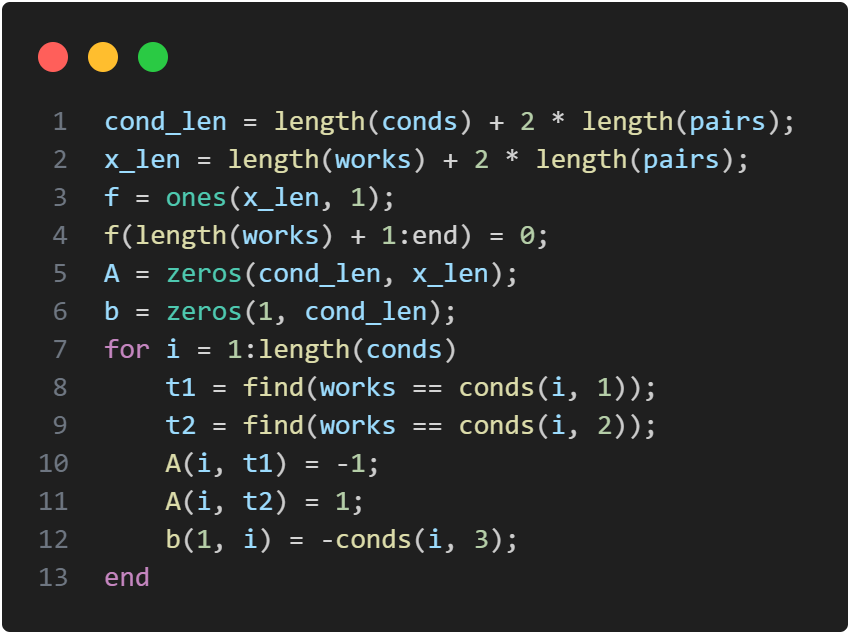
Таким образом желтый цвет – первый исполнитель (2 задачи), а синий – второй (3 задачи), что дает нам дополнительных бинарных переменных и

дополнительных ограничений:

После чего решим задачу Matlab с использованием функции intlinprog. Для начала зададим наш граф, как делали это ранее, также добавим пары для новых ограничений и длины путей на каждой из задач:



Далее создадим массив, как в пункте 2.1, но используя наши объявления:

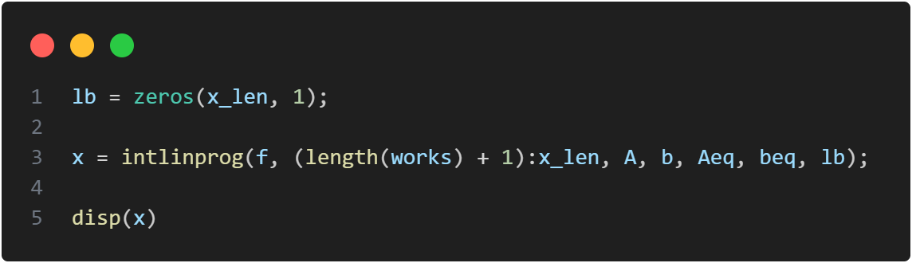


Теперь необходимо создадим уравнения, которые добавились дополнительными ограничениями:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Все необходимые переменные были созданы, теперь запустим intlinprog и посмотрим на результат:



В результате запуска получим следующие значения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Моменты начала работ | |  | Бинарные переменные | |
|  | 5 |  |  | 0 |
|  | 0 |  |  | 1 |
|  | 0 |  |  | 1 |
|  | 0 |  |  | 0 |
|  | 11 |  |  | 1 |
|  | 19 |  |  | 0 |
|  | 18 |  |  | 0 |
|  | 5 |  |  | 1 |
|  | 5 |  |  |  |
|  | 14 |  |  |  |
|  | 14 |  |  |  |
|  | 24 |  |  |  |
|  | 24 |  |  |  |
|  | 33 |  |  |  |
|  | 40 |  |  |  |
|  | 48 |  |  |  |

Табл. 2.3. Результат работы программы.

Как мы видим по значениям бинарных переменных и времени начала работы наши условия выполняются.

Первый исполнитель выполняет сначала работу 14, а потом 12 (т. к. значение =5, а =0)

Второй исполнитель начинает с работы 16 (т. к. это единственная доступная со старта, что видно по нулевому значению ), далее решается работа 56 (т. к. до неё первой доходит очередь), после чего решается работа 36.

Остальные работы считаются параллельно, что видно, например по равным временам начала выполнения у работ 56 и 58 или 67 и 68.

## Найти характеристики , и расписания выполнения комплекса работ с использованием метода динамического программирования. Привести соответствующие уравнения Беллмана. Определить критические пути на графе.

Каждому узлу на графе можно сопоставить два момента: минимальное время, когда событие будет осуществлено и наиболее поздний момент .

Воспользуемся методом динамического программирования и определим наиболее ранние моменты для каждого узла графа:

Изображение выглядит как линия, диаграмма, круг

Автоматически созданное описание

Рис. 2.3. Граф.

Самое ранее время начала выполнения работы – время, когда выполнятся все работы, предшествующие заданной:

Полученные значение совпадают с полученными в предыдущих пунктах, что свидетельствует о корректности проделанной работы.

Теперь используя полученные значения определим наиболее поздние моменты времени :

По формуле определим резервы времени выполнения всех работ:

Работы, у которых резерв равен 0 – критический путь. Их длительность напрямую влияет на продолжительность выполнения всех работ.

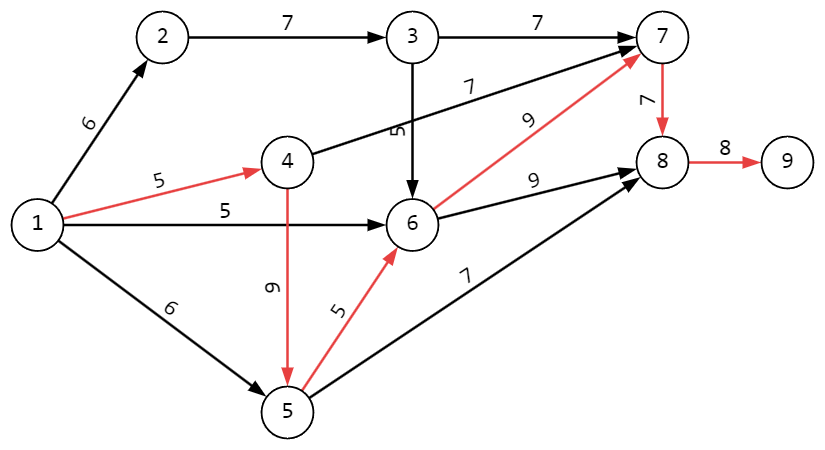


Рис. 2.4. Критический путь в графе.

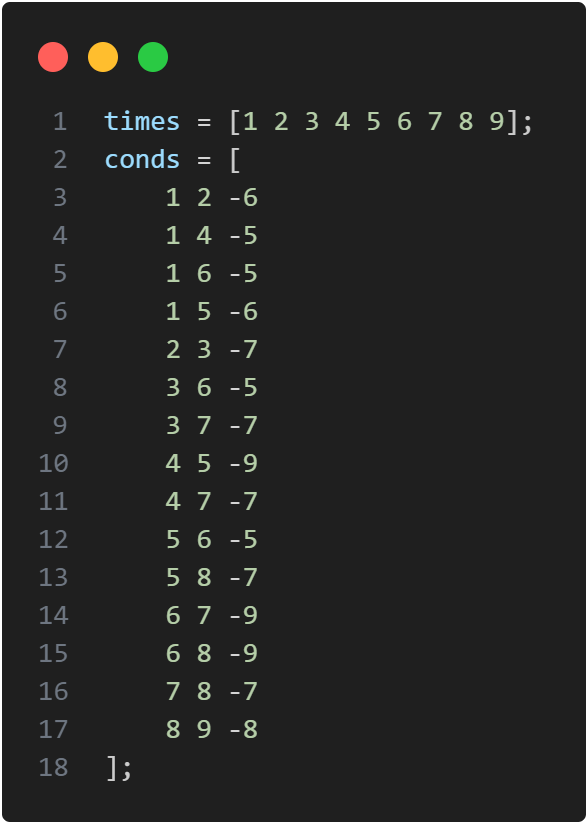
## Найти те же характеристики , и расписания выполнения комплекса работ с использованием математического программирования.

Оптимизационная задача для поиска наиболее ранних моментов может быть сформулирована следующим образом:

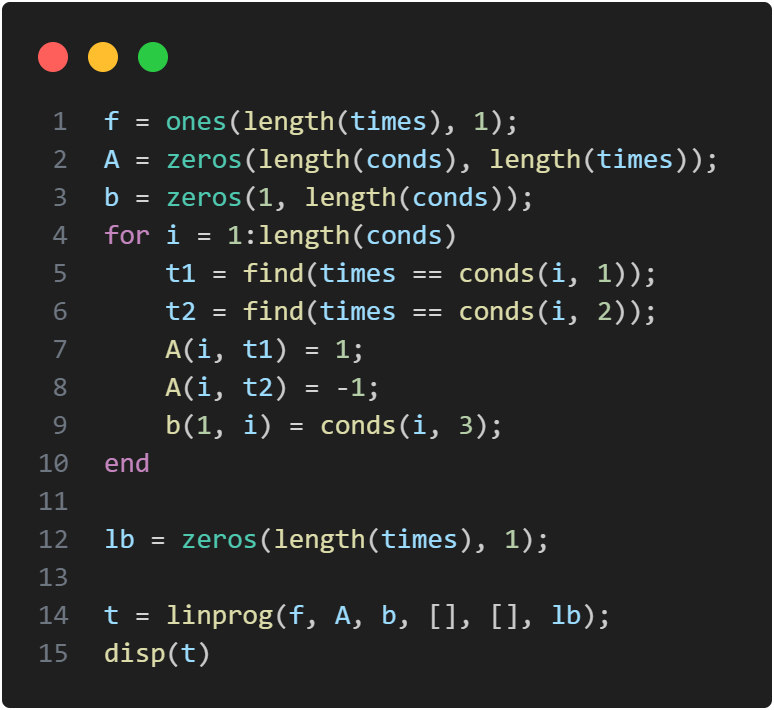
Для исходного графа получим следующую оптимизационную задачу:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Запишем эти выражения в массивы:



Запишем эти выражения в виде, подходящем для linprog и вычислим:



Полученный результат выглядит следующим образом:

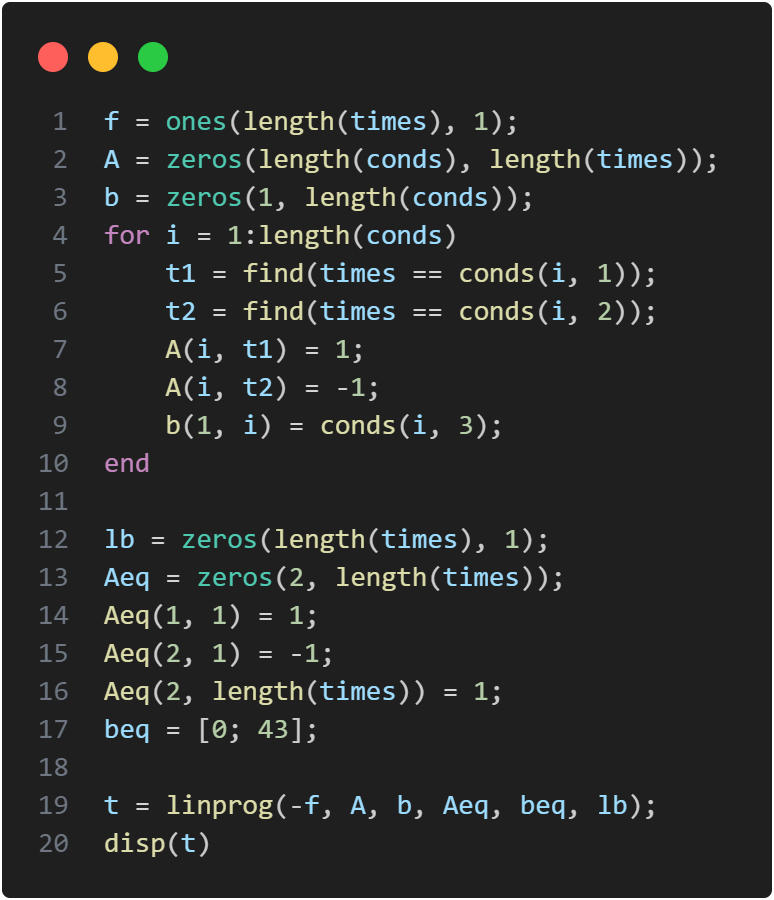
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 0 | 6 | 13 | 5 | 14 | 19 | 28 | 35 | 43 |

Табл. 2.4. Результат решения программой.

Как можно заметить, они идентичны полученным ранее другим способом.

Оптимизационная задача для поиска наиболее поздних моментов может быть сформулирована следующим образом:

Для решения поставленной задачи необходимо чуть-чуть изменить вызов функции, объявления условий останется аналогичным:



Результатом выполнения будет следующим:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 0 | 7 | 14 | 5 | 14 | 19 | 28 | 35 | 43 |

Табл. 2.5. Результат выполнения программы.

Как можно заметить, эти значения идентичны посчитанным ранее.

Очевидно, что значение будет аналогично равно посчитанному ранее.

## Определить помимо полных резервов времени работ резервы времени, относящиеся к событиям сетевого графа, а именно .

# Вывод

# Ссылки

# Приложение

Листинг на github: <https://github.com/DafterT/SADM_6_1>