

Санкт-Петербургский политехнический университет  
Петра Великого

Физико-механический институт

Кафедра «Прикладная математика»

**Отчёт по лабораторной работе  
по дисциплине «Компьютерные сети»  
Задача византийских генералов**

Выполнил студент:  
Габдрахманов Булат Маратович  
группа: 5040102/20201

Проверил:  
к.ф.-м.н., доцент  
Баженов Александр Николаевич

Санкт-Петербург  
2023 г.

## Содержание

1	Постановка задачи	2
2	Теория	2
3	Реализация	3
4	Результаты	3
5	Заключение	5

## 1 Постановка задачи

Необходимо разработать алгоритм для решения задачи византийских генералов.

## 2 Теория

Существует группа из  $n$  генералов, среди которых  $t$  являются предателями. Каждый из этих  $n$  генералов обладает надежным каналом связи, который исключает возможность подмены сообщений. Все лояльные генералы, их количество составляет  $n - t$ , всегда отправляют правдивые и неизменные сообщения. В то же время, каждый из  $t$  предателей отправляет неправдивые и потенциально изменяемые сообщения. Задача лояльных генералов заключается в идентификации предателей посредством обмена сообщениями.

Рассмотрим решение этой проблемы в специфическом случае, когда количество предателей остается постоянным и  $n > 3t$ . Для такой ситуации применим алгоритм Лампорта, который включает в себя определенные этапы.

- Каждый из генералов отправляет сообщение всем остальным: верные генералы отправляют правдивое сообщение, а предатели – неправдивое.
- В результате, у каждого генерала образуется массив из  $n$  элементов, который включает полученные сообщения и собственное.
- После этого каждый генерал пересылает остальным сформированный массив, полученный на предыдущем этапе.
- В финальной стадии каждый генерал имеет серию векторов: свой собственный и те, что были получены от других генералов. Для каждого  $i$ -го элемента в этих векторах определяется наиболее часто встречающееся значение. Если оно встречается как минимум  $n - t$  раз, его считают истинным и включают в итоговый вектор, в противном случае в итоговый вектор добавляется ноль.

Таким образом, все верные генералы в итоге получают идентичный набор значений.

### 3 Реализация

Весь код написан на языке Python (версии 3.9). Каждый генерал работает в отдельном потоке. Предполагается, что отправленное сообщение не может быть потерянно. [Ссылка на GitHub с исходным кодом](#).

### 4 Результаты

Для демонстрации работы алгоритма используем пример, где общее количество генералов составляет  $n = 9$ , среди которых  $m = 2$  являются предателями. Верные генералы отправляют сообщения, содержащие их порядковые номера (индексы), в то время как предатели отправляют сообщения формата  $tr\_i$ , где  $i$  — порядковый номер генерала-предателя. Предателями являются генералы под номерами 7 и 8.

В ходе первых двух этапов работы алгоритма каждый генерал получает следующий вектор значений.

```
Первый вектор генерала 8: 0, 1, 2, 3, 4, 5, 6, tr_8,  
Первый вектор генерала 3: 0, 1, 2, 4, 5, 6, tr_3, tr_3,  
Первый вектор генерала 0: 1, 2, 3, 4, 5, 6, tr_0, tr_0,  
Первый вектор генерала 7: 0, 1, 2, 3, 4, 5, 6, tr_7,  
Первый вектор генерала 1: 0, 2, 3, 4, 5, 6, tr_1, tr_1,  
Первый вектор генерала 5: 0, 1, 2, 3, 4, 6, tr_5, tr_5,  
Первый вектор генерала 4: 0, 1, 2, 3, 5, 6, tr_4, tr_4,  
Первый вектор генерала 2: 0, 1, 3, 4, 5, 6, tr_2, tr_2,  
Первый вектор генерала 6: 0, 1, 2, 3, 4, 5, tr_6, tr_6,
```

Рис. 1: Сформированные векторы у каждого генерала

На следующем этапе каждый верный генерал отправляет ранее сформированный им вектор другим генералам. Предатель же, посылая сообщения, вносит изменения в свой вектор: к каждому элементу добавляется фраза  $tr\_i$ , где  $i$  обозначает порядковый номер генерала, которому отправляется сообщение.

В результате этого шага у каждого генерала складывается собственный набор векторов. Для наглядности рассмотрим по одному примеру вектора, полученного верным генералом и предателем.

```

id генерала = 1
Полученный вектор от генерала 1: : 0, 2, 3, 4, 5, 6, tr_1, tr_1,
Полученный вектор от генерала 8: : tr_1_0, tr_1_1, tr_1_2, tr_1_3, tr_1_4, tr_1_5, tr_1_6, tr_1_tr_8,
Полученный вектор от генерала 6: : 0, 1, 2, 3, 4, 5, tr_6, tr_6,
Полученный вектор от генерала 7: : tr_1_0, tr_1_1, tr_1_2, tr_1_3, tr_1_4, tr_1_5, tr_1_6, tr_1_tr_7,
Полученный вектор от генерала 0: : 1, 2, 3, 4, 5, 6, tr_0, tr_0,
Полученный вектор от генерала 3: : 0, 1, 2, 4, 5, 6, tr_3, tr_3,
Полученный вектор от генерала 5: : 0, 1, 2, 3, 4, 6, tr_5, tr_5,
Полученный вектор от генерала 4: : 0, 1, 2, 3, 5, 6, tr_4, tr_4,
Полученный вектор от генерала 2: : 0, 1, 3, 4, 5, 6, tr_2, tr_2,
id генерала = 7
Полученный вектор от генерала 7: : 0, 1, 2, 3, 4, 5, 6, tr_7,
Полученный вектор от генерала 8: : tr_7_0, tr_7_1, tr_7_2, tr_7_3, tr_7_4, tr_7_5, tr_7_6, tr_7_tr_8,
Полученный вектор от генерала 6: : 0, 1, 2, 3, 4, 5, tr_6, tr_6,
Полученный вектор от генерала 0: : 1, 2, 3, 4, 5, 6, tr_0, tr_0,
Полученный вектор от генерала 3: : 0, 1, 2, 4, 5, 6, tr_3, tr_3,
Полученный вектор от генерала 5: : 0, 1, 2, 3, 4, 6, tr_5, tr_5,
Полученный вектор от генерала 4: : 0, 1, 2, 3, 5, 6, tr_4, tr_4,
Полученный вектор от генерала 2: : 0, 1, 3, 4, 5, 6, tr_2, tr_2,
Полученный вектор от генерала 1: : 0, 2, 3, 4, 5, 6, tr_1, tr_1,

```

Рис. 2: Сформированный набор векторов для верного генерала (индекс 0) и предателя (1)

После завершения всех этапов алгоритма, каждый генерал формирует результирующий вектор. В этом векторе элемент None обозначает генерала-предателя.

Таким образом, результирующий вектор каждого генерала содержит либо порядковые номера верных генералов, либо None на месте предателей.

```

Результирующий вектор для генерала 0: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 8: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 3: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 7: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 6: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 4: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 5: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 2: [0, 1, 2, 3, 4, 5, 6, None, None]
Результирующий вектор для генерала 1: [0, 1, 2, 3, 4, 5, 6, None, None]

```

Рис. 3: Итоговый набор у каждого генерала

Как видно, в сценарии, когда все генералы сформировали одинаковый итоговый вектор, это означает, что они пришли к согласию и успешно определили предателей. Это демонстрирует эффективность алгоритма в случае, когда  $n > 3m$ .

Теперь рассмотрим ситуацию, когда  $n \leq 3m$ . Возьмем для примера  $n = 5$  и  $m = 2$ . В этом случае, после завершения всех шагов алгоритма, каждый генерал сформировал следующий итоговый вектор:

```
Результирующий вектор для генерала 3: [0, 1, 2, None, None]
Результирующий вектор для генерала 4: [0, 1, 2, None, None]
Результирующий вектор для генерала 1: [0, 'tr_1_1', 'tr_1_2', None, None]
Результирующий вектор для генерала 0: [0, 1, 2, None, None]
Результирующий вектор для генерала 2: [0, 1, 'tr_2_2', None, None]
```

Рис. 4: Итоговый набор у каждого генерала в случае  $n \leq 3m$

Как видно на рисунке 4, верные генералы не смогли прийти к согласию. Это указывает на ограничения алгоритма в условиях, когда число предателей сопоставимо или превышает одну треть от общего числа генералов.

## 5 Заключение

Из представленных результатов становится очевидно, что когда условие  $n \leq 3m$  выполняется, существует риск того, что генералы не смогут достичь консенсуса. Это связано с тем, что в таких ситуациях влияние предателей становится слишком значительным, что мешает достижению единого мнения среди верных генералов.

В то же время, в случаях, когда  $n > 3m$ , алгоритм обеспечивает гарантию того, что согласие всегда может быть достигнуто. Это происходит благодаря тому, что количество верных генералов достаточно велико по сравнению с количеством предателей, что позволяет надежно идентифицировать ложные сообщения и достигать единого решения.