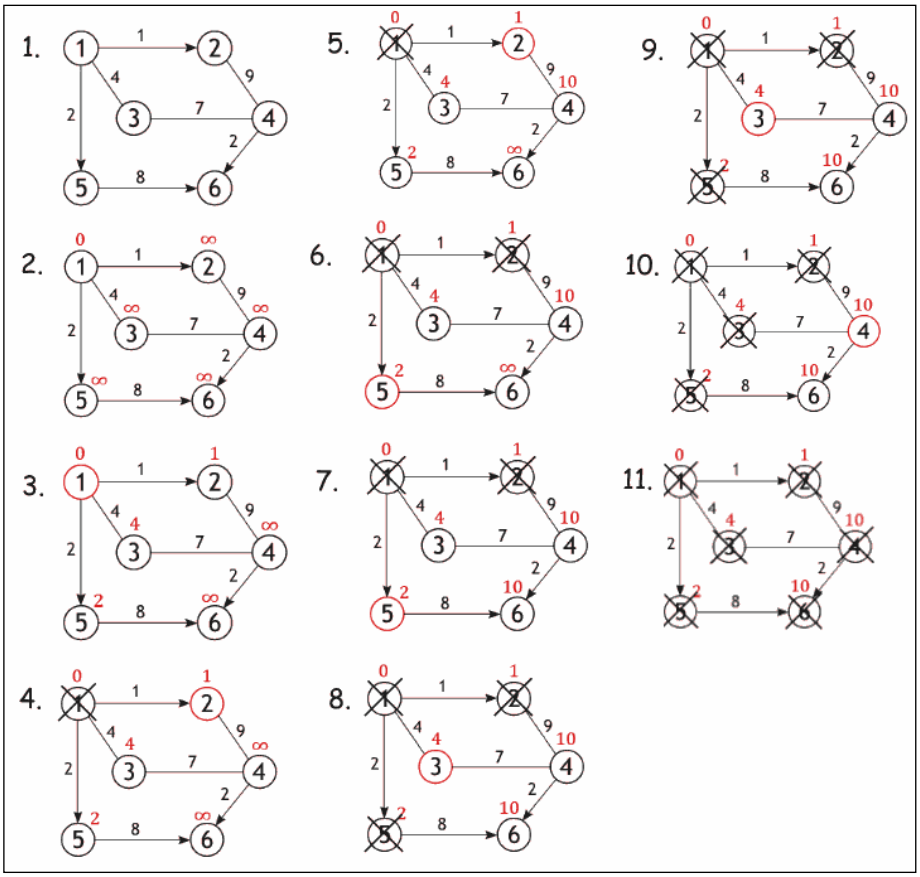

Малюнок 5. Приклад роботи алгоритма Дейкстри

Головний недолік алгоритму в тому, що його не можна використовувати для пошуку найкоротшої відстані у графах із негативними ребрами. Стратегія алгоритму завжди вибирає негативне число як меншу відстань.

Список літературних джерел

1. *Data Structure & Algorithms - Spanning Tree URL:* https://www.tutorialspoint.com/data_structures_algorithms/spanning_tree.htm
2. *Алгоритм Краскала URL:* https://www.wikiwand.com/ru/Алгоритм_Краскала
3. https://kvodo.ru/dijkstra-algorithm.htm

УДК 004.01

Суліма В.К., студент 1 курсу
спеціальності 122 «Комп'ютерні науки»
Ніколюк П.К., професор
кафедри інформаційних технологій

ПАРАДОКС ДНІВ НАРОДЖЕННЯ ТА ЙОГО ЗАСТОСУВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Парадокс днів народження – твердження, що у теорії ймовірностей, оцінює шанс того, що у випадково зібраній групі осіб, хоча б в одної пари

збігаються дні народження(день та місяць). В групах, де кількість випадково вибраних людей більш як 23, ймовірність збігу днів народження понад 50%, що суперечить інтуїтивній уяві, щодо ситуації, у більшості людей.[1]

Основа даної задачі, доволі часто використовується при поясненні та аналізі інших задач з теорії ймовірностей, криптографії, математичній статистиці, де використовується рандомізація, також її використовують при аналізі хеш-функцій. На разі існує багато видозмінених варіацій цього завдання.

Розглянемо розв'язання цієї задачі при наявності 23 людей – припускаємо, що ймовірність того, що дві людини народились в один день P(A), тоді ймовірність того, що дві людини народились в різні дні P(B). Через те, що події взаємо заперечні, то:

$$P(A) = 1 - P(B)$$

Отже, знайдемо ймовірність, що всі учасники народились в різні дні, перший може народитись в будь-який з 365 днів, другий може народитись в будь-який з 364 днів, що лишились, продовжуючи цей ланцюжок отримуємо:

$$P(B) = \frac{365}{365} \times \frac{364}{365} \cdots \times \frac{337}{365} \times \frac{336}{365} \approx 0{,}492$$

Звідси ж:

$$P(A) = 1 - 0{,}492 = 0{,}508 = 50{,}8\%$$

Використовуючи комбінаторний аналіз для P(B) можна вивести загальну формулу, для будь-якої кількості людей, якщо кількість випадків без повторень(для чого використовується комбінація розміщень без повторень) розділити на загальну кількість можливих варіацій(для чого використовується комбінація розміщень з повтореннями):

$$P(B) = \frac{365!}{(365 - n)!} \times \left(\frac{1}{365}\right)^n$$

n – кількість людей в групі

Зі збільшенням кількості осіб, буде збільшуватись ймовірність того, що принаймні двоє народились в один день, поки не досягне 366, при якому ймовірність події буде 100%, в задачі розглядається невисокосний рік.

Таблиця 1

Приблизна ймовірність для різної кількості людей

n	**P(n)**
10	11.7 %
20	41.1 %
23	50.7 %
30	70.6 %
50	97.0 %
57	99.0 %
100	99.99997 %
300	$(100 - (6 \times 10^{-80}))$ %
≥366	100 %

Отже, дана задача не є парадоксом, а має математичне розв’язання, а назву парадоксу, вона отримала через примарну нелогічність, що всього 23 людей вистачає для 50% вірогідності заданої події, оскільки більшість людей припускає, що це число має бути на рівні 183.

Як вже згадувалося, парадокс використовують у хеш-функціях - це будь-яка функція, яку можна використовувати для перетворення даних довільного розміру у значення фіксованого розміру, наприклад:[2]

- H("привіт") = 4a;
- H("добре") = c9;
- H("можливо") = 1d;
- H("припустимо") = c9.

Через те, що всі хеш-коди мають однакову довжину(у різних хеш-функцій довжина хеш-коду може відрізнятись), то вони все одно мають повторення, оскільки обсяг інформації, який можна через них пропустити перевищує кількість комбінацій у хеш-кодах, що призводить до присвоєння різним даним однакових хеш-кодів, що називається колізією хешів. Це означає, що так само як знаходили двох людей з однаковими датами дня народження, ми так само можемо знайти два повідомлення, що мають однаковий хеш-код.

Небезпечність колізії можна побачити на прикладі паролів – при реєстрації до пароля, який вводить користувач застосовується хеш-функція результат якої зберігається в БД, одже коли злочинець отримає доступ до бази даних він не зможе дізнатись оригінальний пароль користувача, але якщо він вміє знаходити колізії, то зможе знайти неоригінальний пароль з тим самим хеш-кодом, таж сама система використовується для підробки електроних підписів.

Кількість можливих значень у різних хеш-функціях:

- MD5 - 2^128 хеш-значень;
- SHA-1 - 2^160 хеш-значень;
- bcrypt - 2^192 хеш-значення.

Узагальнюючи формулу задачі, де замість кількості днів у році може бути вибірковий простору(H), а замість людей набір данних(N):

$$P(H) = \frac{H!}{(H-N)!} \times \left(\frac{1}{H}\right)^N$$

Якщо пропустити через це рівняння вибіркові простори різних хеш-функцій, то можна побачити, що колізія настає приблизно при проходженні $2^{x/2}$ операцій. Це означає, що 50% шанс знайти колізію SHA-1 (вибірковий простір 2^{160}) приблизно після 2^{80} операцій і т.п. [3]

Для боротьби з таким типом хакерських атак використовують сучасні хеш-функції SHA-256, SHA-512 та подібні, де використання даного способу вже не є доцільним, але з розвитком квантових комп’ютерів та їх обчислювальною здатністю все може змінитись.

Отже, можна зробити висновок, що парадокс днів народження все ще є гарним прикладом для розв'язання подібних задач з теорії ймовірностей,

математичної статистики, та має чудові можливості для пошуку колізій у хеш-функціях, а з розвитком технологій може збільшити свою актуальність.

Список використаної літератури:

1. *Birthday problem URL: https://en.wikipedia.org/wiki/Birthday_problem*
2. *Answering the Birthday Problem in Statistics URL: https://statisticsbyjim.com/fun/birthday-problem/*
3. *What is the birthday paradox and how we can use it in cryptography?https://justcryptography.com/the-birthday-paradox/*

УДК 004.056

Саган М.Я. студент 2 курсу СО «Магістр»
спеціальності 122 «Комп'ютерні науки»
Антонов Ю.С. к. фіз.-мат. н., доцент кафедри інформаційних технологій

ПРОБЛЕМА ЗАХИСТУ NGINX СЕРВЕРІВ ВІД ВІД DDoS АТАК

Донецький національний університет імені Василя Стуса, м Вінниця

Сфера IT розвивається надзвичайно швидкими темпами, нові проекти з'являються кожного дня. Завдяки хмарним провайдерам, кожен може отримати можливість створювати та розгортати свої рішення в короткий час та за розумні гроші. Хмарні сервіси звільняють користувачів від необхідності обслуговування власної інфраструктури, необхідності оплати роботи спеціалістів, адміністраторів, та позбавляють необхідності вивчати предмет самостійно. Все сказане вище значно знижує поріг входу. Хмарні рішення надають надзвичайну гнучкість при виборі ресурсів, але іноді, через це страждають характеристики компонентів. Так диски, які надає амазон, мають дуже низький показник IOPS (операції введення/виведення за секунду).

Разом із цим доволі гостро стає проблема захисту хмарних та класичних ресурсів від зовнішніх атак, які використовують як специфічні дефекти системи для атаки так і більш прості, але не менш дієві DoS\DDoS атаки [1-3].

В представленій роботі пропонується програмне рішення, що дозволяє ефективно обробляти логи веб-сервера NGINX та реагувати на загрози. Продукт мінімально впливає на роботу сервера, в плані споживання його ресурсів та створення затримок. Спроєктовано продукт таким чином, щоб він міг ефективно працювати із повільними дисками, аби уникнути їх використання взагалі. Джерелом логів є веб сервер NGINX [4], так як він являє собою швидкий і максимально продуктивний програмний продукт, який прекрасно працює на широкому спектрі засобів, та може обробляти великі