

2. What is a database. *amazon.com*. URL: <https://aws.amazon.com/what-is/database/> (дата звернення: 15.05.2024).
3. What is data model. *Center for Data, Analytics and Reporting (CeDAR)*. URL: <https://cedar.princeton.edu/understanding-data/what-data-model> (дата звернення: 15.05.2024).
4. What is SQL. *geeksforgeeks.org*. 2024. URL: <https://www.geeksforgeeks.org/what-is-sql/> (дата звернення: 15.05.2024).
5. Gulati A. Real-time data analytics. *upGrad. Knowledgehut*. 24 Apr, 2024. URL: <https://www.knowledgehut.com/blog/data-science/real-time-data-analytics> (дата звернення: 15.05.2024).
6. Real-Time Data Processing. *Gigaspaces*. URL: <https://www.gigaspaces.com/data-terms/real-time-data-processing> (дата звернення: 15.05.2024).

УДК 004.9

Грибаніна А. О., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Поремський Ю. В., канд. техн. наук,
 старший викладач кафедри інформаційних технологій

ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ ЧЕРГИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Черга – це змінюваний упорядкований (чи ні) набір елементів. Додавання елементу в чергу проводиться з одного кінця (хвоста черги, REAR), а вибірка – з іншого кінця (голови черги, FRONT) відповідно до правила «Першим прийшов – першим пішов» (FIFO: First Input – First Output). Така черга є простою чергою без пріоритетів (рис. 1). Часто використовуються черги з пріоритетами, в них більш пріоритетні елементи включаються ближче до голови черги, вибірка здійснюється зазвичай з голови черги.

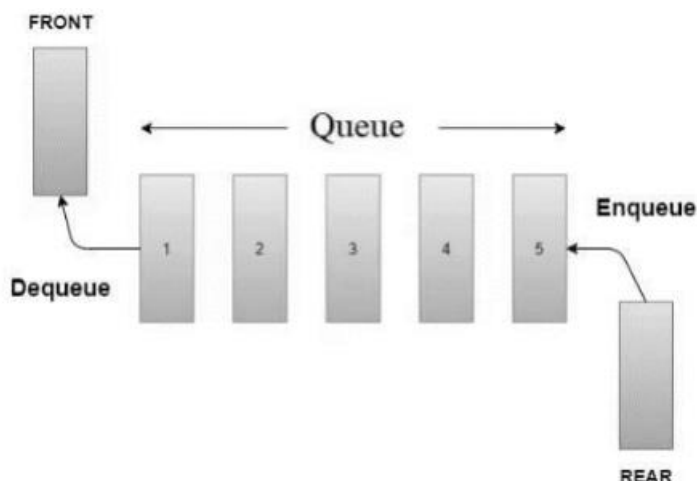


Рис. 1. Схема простої черги

Основні операції над чергою – ті ж, що і над стеком – включення, виключення, визначення розміру. Класична реалізація черги вимагає реалізувати структуру даних із такими операціями:

1. Створення порожньої черги.

2. Операція `empty()` – визначення, чи є черга порожньою. Повертає булеве значення.

3. Операція `enqueue(item)` – додати елемент `item` до кінця черги.

4. Операція `dequeue()` – взяти елемент з початку черги.

Як і у випадку зі стеком, крім зазначених вище операцій, опціонально визначають інші операції: перегляд елемента в голові та хвості черги без їх видалення з черги, розмір черги, тобто кількість елементів у ній, тощо.

Реалізація черги у вигляді масиву ефективна тоді, коли максимальний розмір черги маленький або відомий заздалегідь. Але якщо розмір масиву під чергу не може бути визначено заздалегідь (а таких випадків більшість), іншої альтернативи, ніж використання представлення черги у вигляді зв'язного списку, не існує.

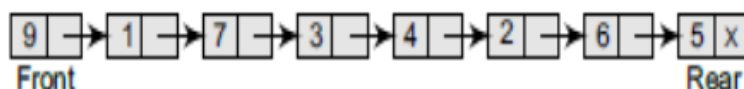


Рис. 2. Реалізація простої зв'язної черги

Основними операціями над чергою є:

- 1) створення і видалення черги;
- 2) включення в чергу нового елемента;
- 3) вибірка елемента з черги.

Під час їх виконання використовуються допоміжні операції, як-от перевірка наявності елементів у черзі і перевірка переповнення черги.

Як і у випадку зі стеком, створення черги зводиться до побудови тільки дескриптора черги. У ньому наявні покажчик на голову черги `front`, покажчик на хвіст черги `rear` і поточний розмір черги `size`. Покажчики `front` і `rear` встановлюються в `NULL`, поле `size` встановлюється в нуль, позначаючи, що черга порожня.

Зводиться до звільнення динамічної пам'яті, отриманої в процесі створення і обробки черги. Перед видаленням дескриптора черги треба видалити всі елементи черги по черзі, з голови черги до її хвоста. В іншому випадку будуть виникати витоки пам'яті.

У разі черги спискової структури переповнення зазвичай не буває, однак перевірка необхідна. Включення в чергу зводиться до отримання динамічної пам'яті під новий елемент, занесення в нього даних і додавання елемента у хвіст черги. Очевидно, що в покажчик наступного елемента черги в старому хвості заноситься значення покажчика на новостворений елемент (новий хвіст черги).

Операція можлива тільки тоді, коли черга не порожня. Вона означає, що вибирається значення з елемента, який знаходиться в голові черги, а сам елемент виключається з черги. Покажчик на голову черги в дескрипторі замінюється на адресу наступного елемента, а пам'ять під колишню голову черги звільняється.

Чергу з пріоритетом можна розглядати як модифіковану чергу, в якій для обслуговування вибирається перший елемент із найвищим пріоритетом із головної частини черги. Сам пріоритет елемента може бути встановлений на основі різних факторів. Черги пріоритетів широко використовуються в операційних системах для виконання процесів із найвищим пріоритетом насамперед. Пріоритет процесу може бути встановлений на основі необхідного процесорного часу для повного завершення процесу. Наприклад, якщо є три процеси, причому першому

процесу необхідно 5 мкс, щоб завершитися, другому процесу необхідно 4 мкс, а третій процес потребує 7 мкс, то другий процес матиме найвищий пріоритет, а отже, буде взятий із черги завдань і виконаний першим.



Рис. 3. Приклад реалізації черги з пріоритетом

Отже, черги з пріоритетом можуть застосовуватись для реалізації алгоритмів паралельної обробки з метою найбільш ефективного використання машинного часу.

Список використаних джерел

1. Грудзинський Ю. Є. Алгоритми та структури даних: навчальний посібник. 2022. URL: <https://ela.kpi.ua/server/api/core/bitstreams/0db974f9-16fa-459c-9f19-fab0021222ed/content>
2. Крєневич А. П. Алгоритми і структури даних. Підручник. Київ: ВПЦ «Київський Університет», 2021. 200 с. URL: <https://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf>
3. Програмна реалізація та дослідження алгоритмів паралельного швидкого сортування / В. О. Денисюк, Н. А. Потапова, О. В. Зелінська, М. Б. Тарасюк. *Вісник Хмельницького національного університету. Технічні науки*. 2023. № 4. С. 95–105. URL: http://journals.khnu.km.ua/vestnik/?page_id=41

УДК 004.1

*Вишневський А. В., здобувач
1 курсу ОС «Магістр»
спеціальності 122 Комп'ютерні науки,
Горяшин А. С., асистент
кафедри інформаційних технологій*

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПРОГНОЗУВАННЯ ЗАБРУДНЕННЯ ПОВІТРЯ ЗА ДОПОМОГОЮ АНАЛІЗУ МЕТЕОРОЛОГІЧНИХ ТА ЕКОЛОГІЧНИХ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі проблема забруднення повітря стає дедалі актуальнішою. Викиди шкідливих речовин від промислових підприємств, транспорту та інших джерел негативно впливають на здоров'я людей і стан навколишнього середовища. Впровадження інформаційних систем для моніторингу та прогнозування рівня забруднення повітря є необхідним кроком для вирішення цієї проблеми.

Інформаційні системи для прогнозування забруднення повітря дають змогу вчасно виявляти та реагувати на критичні ситуації, що може значно знизити негативний вплив на здоров'я населення та довкілля. Використання сучасних інформаційних технологій та аналізу великих обсягів даних забезпечує підвищення точності та ефективності прогнозів.

Останні дослідження показують, що аналіз метеорологічних даних у поєднанні з даними про забруднення повітря значно покращує точність моделей прог-