

Семинар 3 по СДП

Опашка. Основни операции. Последователна и свързана реализация. Приложения

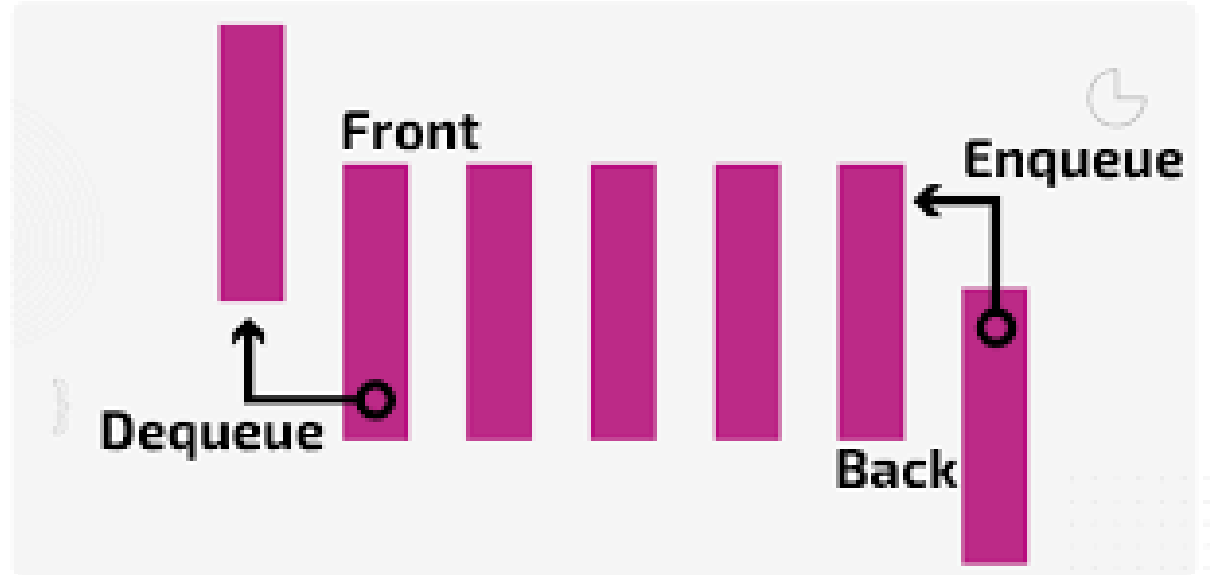


Що е опашка

- Структура от данни, в която данните се добавят отзад, а се взимат отпред.
- Изпълнява условието FIFO (First In First Out)
- Като опашка в магазин (не балканската версия)

Основни операции

- Enqueue(push)
- Dequeue(pop)
- Peak (front)
- Size
- IsEmpty



Реализации

- Последователна
- Свързана

Приложения

- Подреждане на някакви задачи
- Опашка за достъп до нещо.
- BFS (по-късно в курса)

Задачи

- Проверете дали една опашка е сортирана
- <https://leetcode.com/problems/number-of-recent-calls/description/?envType=problem-list-v2&envId=queue>

В разгорещения сблъсък между ветераните от втората световна война - Доналд Дък и Дядо Байдън, този път не за кой има по-голяма пенсия, а за правото да бъде 46-ти президент на САЩ, те броят бюлетини. От цялата държава прииждат гласове за броене. Бюлетините идват на поточна линия при преброителите. Те обаче са връстници на претендентите и са малкобавнички в броенето, за това трябва да им се създаде система с която да има изчакване на прииждащите бюлетини. Също така след като бъде отчетен глас, той се поставя на една голяма купчина. От тази купчина може да се взима бюлетина само най-отгоре, ако някой от претендентите реши, че иска.

Системата ще трябва да се реализира чрез няколко команди:

> **vote** <Name> <State> <blue/red> - добавя бюлетина на поточната линия.

- **count** - взима бюлетината, която е на ред да се преброи и я поставя в правилната купчина.

- **info-election** - дава процентна информация за изборите до момента.

- **take** <blue/red> - взима най-горната бюлетина от съответната купчина, дава информация за нея и я добавя в статистиката.

Статистиката представлява информация за всеки щат за който е постъпил глас

- **info** <State> - дава хронологична информация за всяка една бюлетина от този щат, без значение от това за кого е гласувано.