Опашка

Трифон Трифонов

Структури от данни и програмиране, спец. Компютърни науки, 2 поток, 2024/25 г.

24 октомври 2024 г.

"Queue" от Xiaojun Deng. CC BY 2.0

АТД: опашка

Хомогенна линейна структура с организация „пръв влязъл — пръв излязъл“ (FIFO)

Операции:

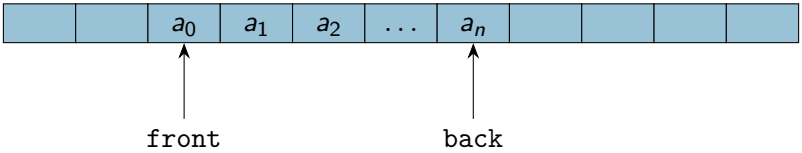
- `create()` — създаване на празна опашка
- `empty()` — проверка за празнота на опашка
- `enqueue(x)` — включване на елемент в края на опашката
- `dequeue()` — изключване на елемент от началото на опашката
- `head()` — достъп до първия елемент

АТД: опашка

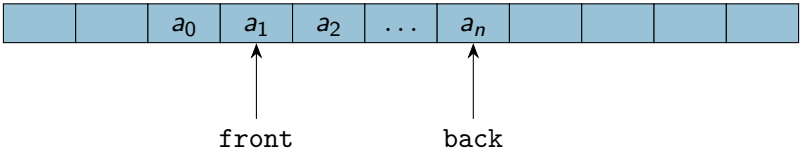
Свойства на операциите

- `create().empty()` = `true`
- `q.enqueue(x).empty()` = `false`
- `create().head()`, `create().dequeue()` — грешка
- `create().enqueue(`x_1`).enqueue(`x_2`)...enqueue(`x_n`).head()` = x_1
- `create().enqueue(`x_1`).enqueue(`x_2`)...enqueue(`x_n`).dequeue()` = `create().enqueue(`x_2`)...enqueue(`x_n`)`

Последователно представяне

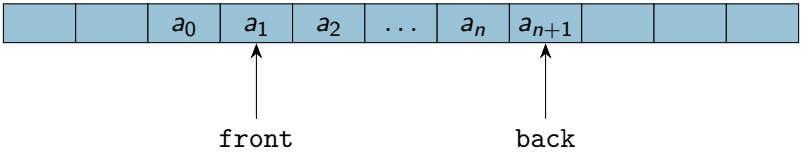


Последователно представяне



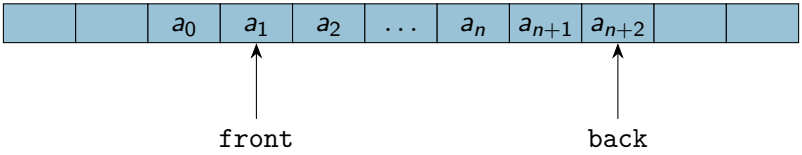
- изключване на елемент (dequeue)

Последователно представяне



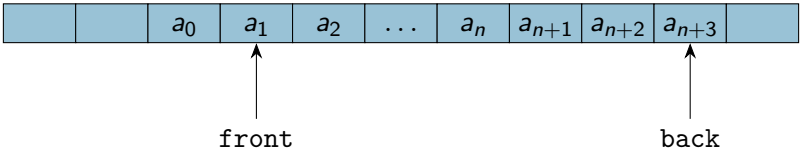
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне



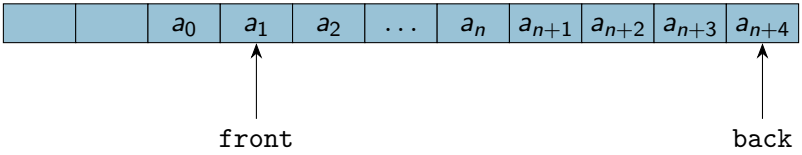
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне



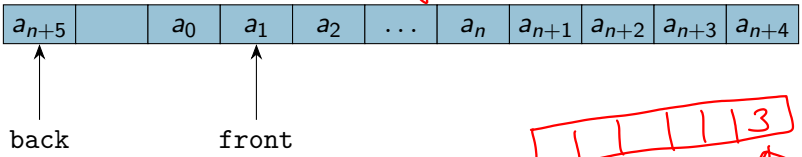
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне



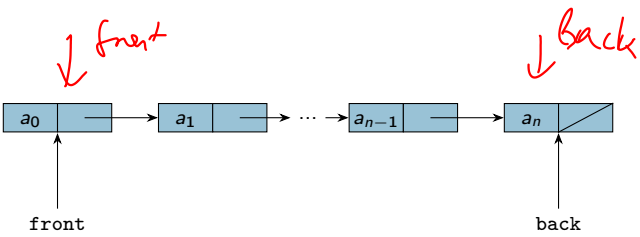
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне

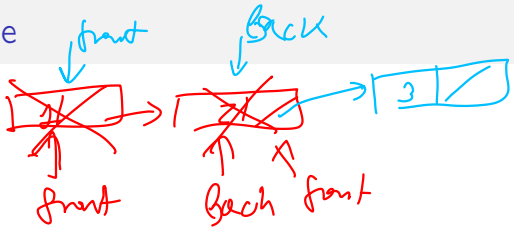


- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Свързано представяне

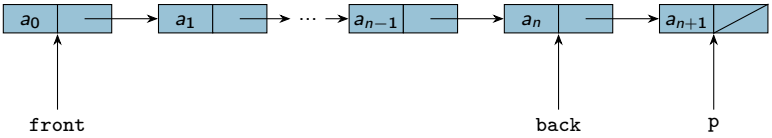


Свързано представяне



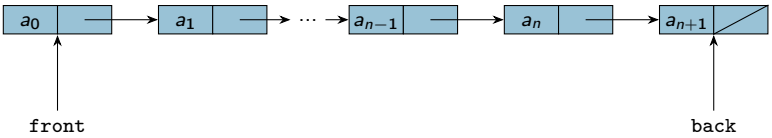
- включване на елемент (enqueue)

Свързано представяне



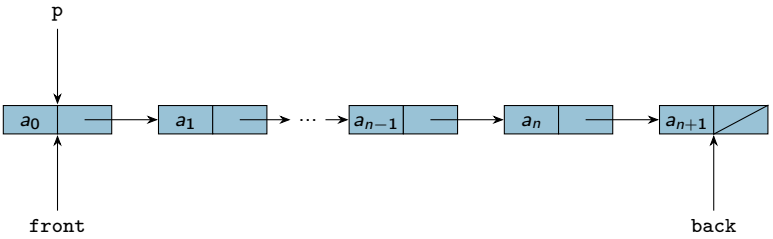
- включване на елемент (enqueue)

Свързано представяне



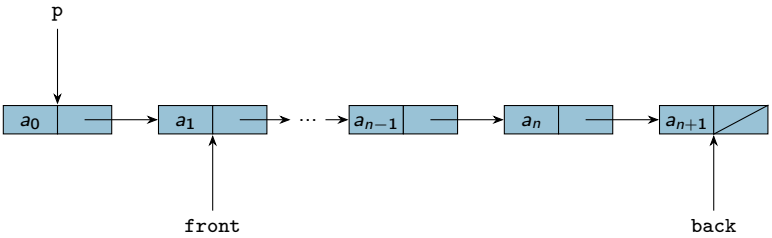
- включване на елемент (enqueue)

Свързано представяне



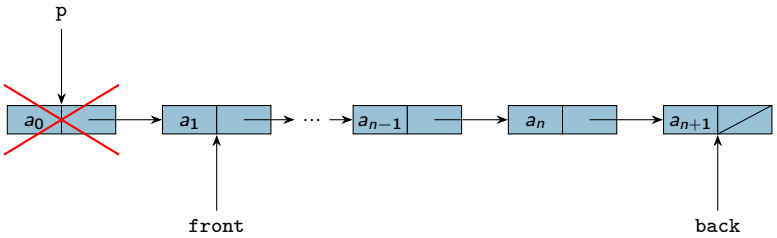
- включване на елемент (enqueue)
- изключване на елемент (dequeue)

Свързано представяне



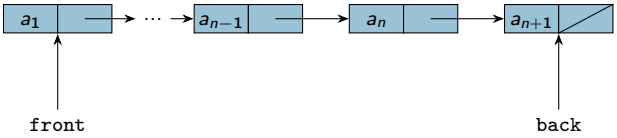
- включване на елемент (enqueue)
- изключване на елемент (dequeue)

Свързано представяне



- включване на елемент (enqueue)
- изключване на елемент (dequeue)

Свързано представяне



- включване на елемент (enqueue)
- изключване на елемент (dequeue)

Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

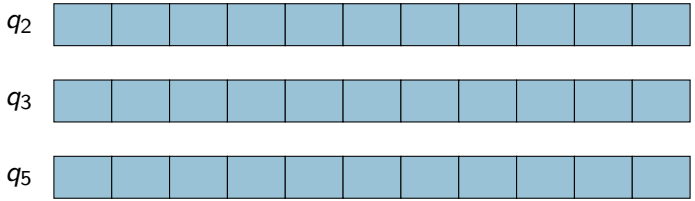
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1

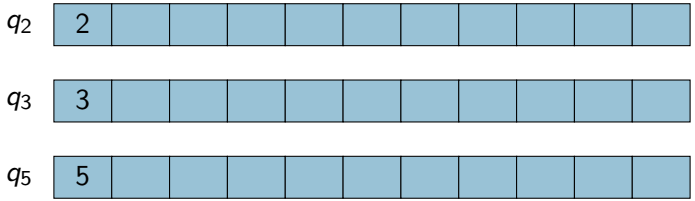
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1

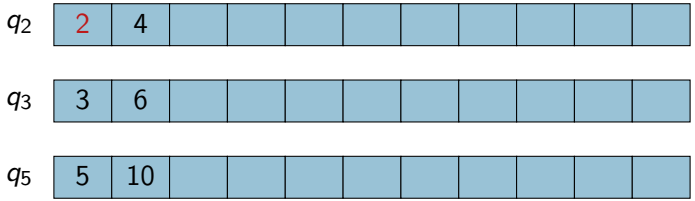
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1

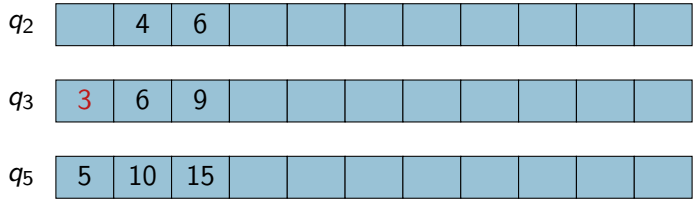
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1, 2

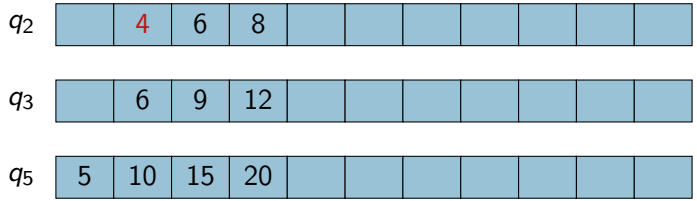
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1, 2, 3

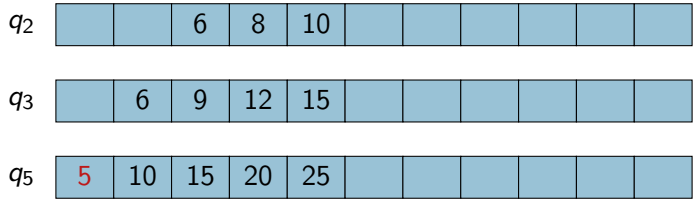
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1, 2, 3, 4

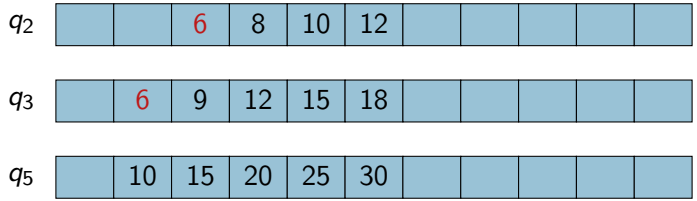
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1, 2, 3, 4, 5

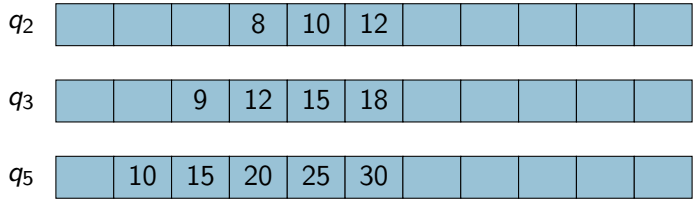
Числа на Hamming

Дефиниция

Казваме, че k е число на Hamming, ако простите делители на k са сред 2, 3 и 5, т.е. $k = 2^x 3^y 5^z$ за $x, y, z \geq 0$.

Задача. Да се изведат в нарастващ ред първите n числа на Hamming.

Решение:



1, 2, 3, 4, 5, 6, ...

Числа на Hamming: коректност

Да се докаже, че:

1. се извеждат **всички** числа на Hamming

Числа на Hamming: коректност

Да се докаже, че:

1. се извеждат **всички** числа на Hamming

Доказателство.

Индукция: $2^x 3^y 5^z$ се извежда, понеже $2^{x-1} 3^y 5^z$ се извежда. □

Числа на Hamming: коректност

Да се докаже, че:

1. се извеждат **всички** числа на Hamming

Доказателство.

Индукция: $2^x 3^y 5^z$ се извежда, понеже $2^{x-1} 3^y 5^z$ се извежда. □

2. се извеждат **само** числа на Hamming

Числа на Hamming: коректност

Да се докаже, че:

1. се извеждат **всички** числа на Hamming

Доказателство.

Индукция: $2^x 3^y 5^z$ се извежда, понеже $2^{x-1} 3^y 5^z$ се извежда. □

2. се извеждат **само** числа на Hamming

Доказателство.

Ако извадим $2^x 3^y 5^z$, в опашките се записват $2^{x+1} 3^y 5^z$, $2^x 3^{y+1} 5^z$, $2^x 3^y 5^{z+1}$. □

Числа на Hamming: коректност

Да се докаже, че:

1. се извеждат **всички** числа на Hamming

Доказателство.

Индукция: $2^x 3^y 5^z$ се извежда, понеже $2^{x-1} 3^y 5^z$ се извежда. □

2. се извеждат **само** числа на Hamming

Доказателство.

Ако извадим $2^x 3^y 5^z$, в опашките се записват $2^{x+1} 3^y 5^z$, $2^x 3^{y+1} 5^z$, $2^x 3^y 5^{z+1}$. □

3. числата на Hamming се извеждат във възходящ ред

Числа на Hamming: коректност

Да се докаже, че:

1. се извеждат **всички** числа на Hamming

Доказателство.

Индукция: $2^x 3^y 5^z$ се извежда, понеже $2^{x-1} 3^y 5^z$ се извежда. □

2. се извеждат **само** числа на Hamming

Доказателство.

Ако извадим $2^x 3^y 5^z$, в опашките се записват $2^{x+1} 3^y 5^z$, $2^x 3^{y+1} 5^z$, $2^x 3^y 5^{z+1}$. □

3. числата на Hamming се извеждат във възходящ ред

Доказателство.

Да допуснем, че на края на някоя опашка добавяме по-малко число. Тогава на предна стъпка трябва да сме добавили по-малко число! □

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

min = 5

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

min = 3

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

$\min = 3$

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

$\min = 1$

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

$\min = 1$

Минимален елемент на опашка

Задача. Дадена е опашка q. Да се изключи от q най-малкият ѝ елемент, като всички останали елементи останат в опашката (не непременно в първоначалния ред).

Решение:

$\min = 1$

Сортиране на опашки с пряка селекция

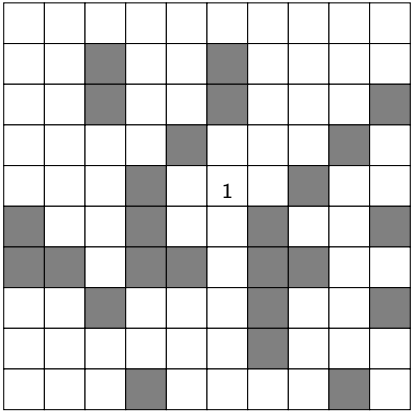
Задача. Да се подредят елементите на опашка в нарастващ ред.

Сортиране на опашки с пряка селекция

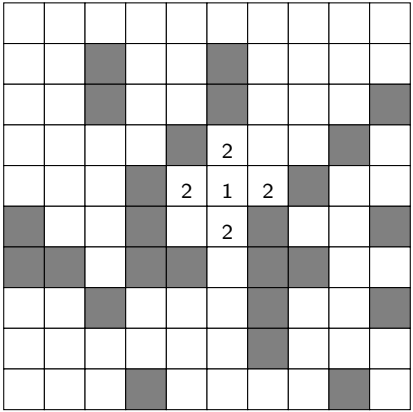
Задача. Да се подредят елементите на опашка в нарастващ ред.

Решение: Използваме нова опашка и прилагаме предната задача върху дадената опашка докато свърши, а минималните елементи поставяме в новата опашка.

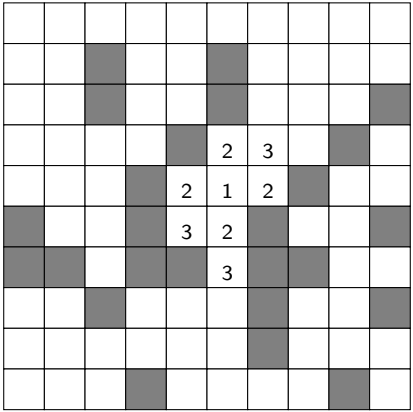
Метод на вълната



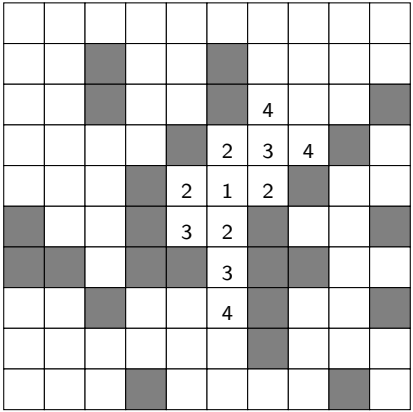
Метод на вълната



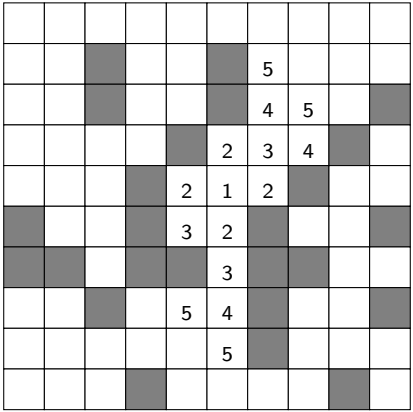
Метод на вълната



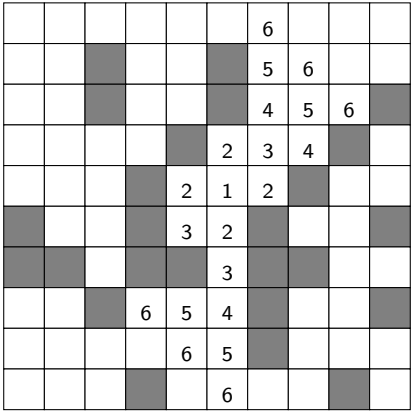
Метод на вълната



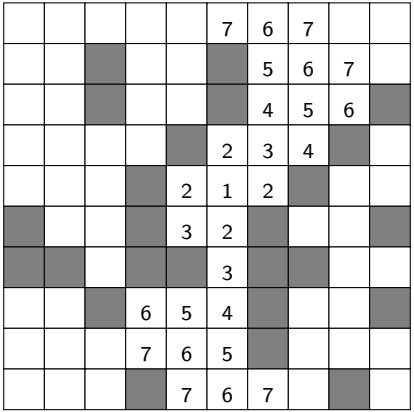
Метод на вълната



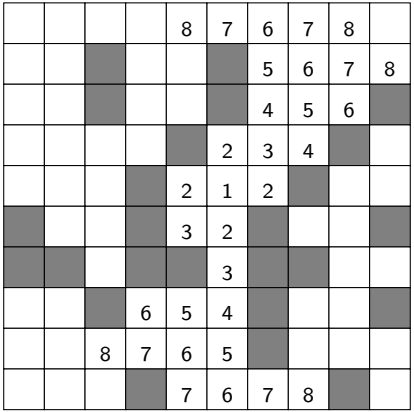
Метод на вълната



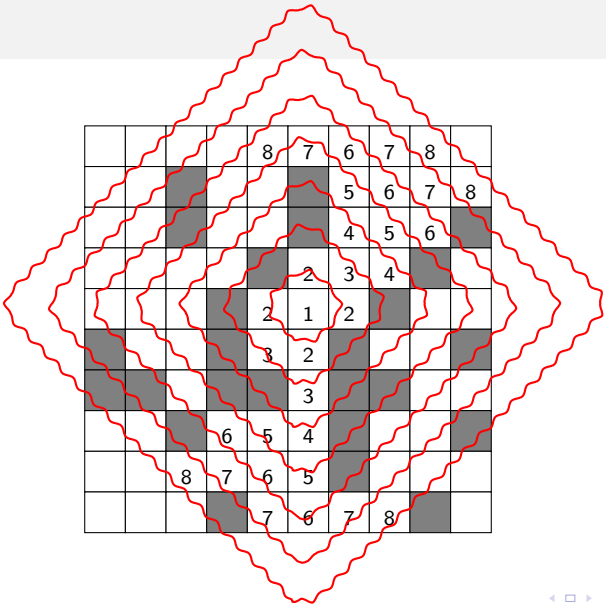
Метод на вълната



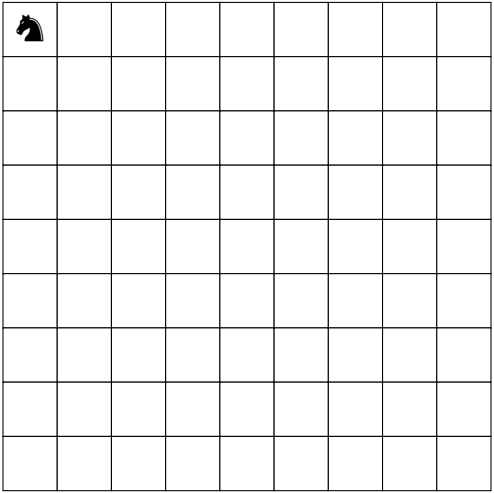
Метод на вълната



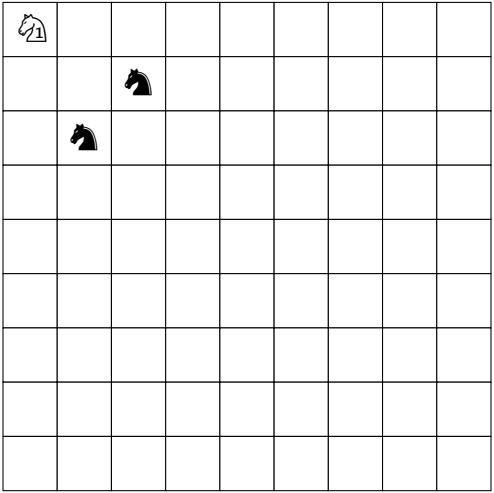
Метод на вълната



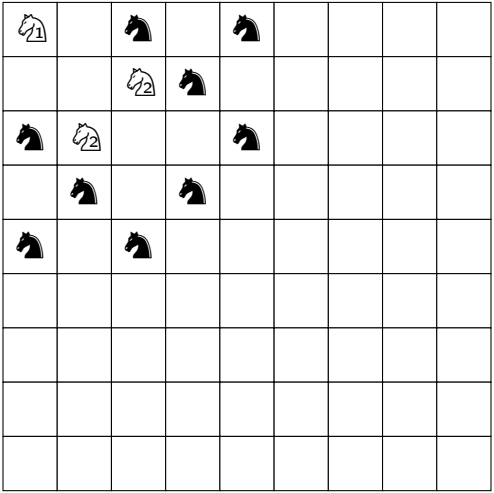
Ход на коня — най-кратък път



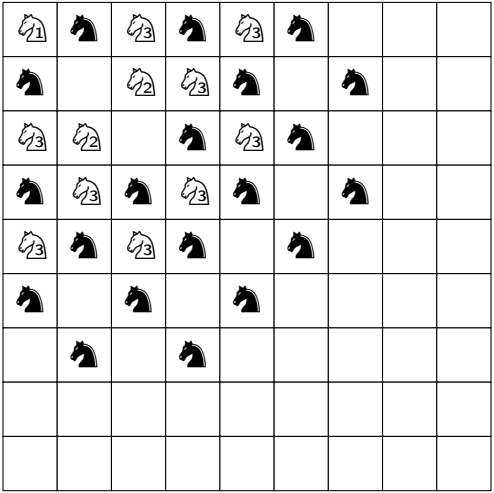
Ход на коня — най-кратък път



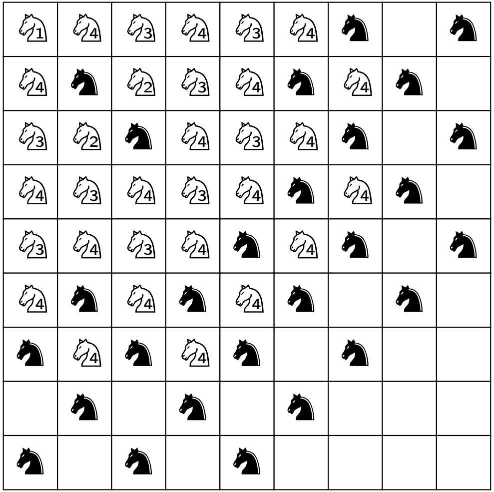
Ход на коня — най-кратък път



Ход на коня — най-кратък път



Ход на коня — най-кратък път



std::queue<T>

- `queue()` — създаване на празна опашка
- `empty()` — проверка за празнота на опашка
- `push(x)` — включване на първи елемент в опашката
- `pop()` — изключване на последен елемент от опашката
- `front()` — първи елемент в опашката
- `back()` — последен елемент в опашката
- `size()` — дължина на опашката
- `==,!=,<,>,<=,>=` — лексикографско сравнение на две опашки