

Опашка

Трифон Трифонов

Структури от данни и програмиране,
спец. Компютърни науки, 1 поток, 2017/18 г.

7 ноември 2017 г.



АТД: опашка

Хомогенна линейна структура с организация “първ влязъл — първ излязъл” (FIFO)

Операции:

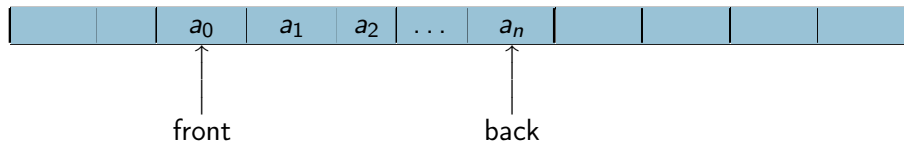
- `create()` — създаване на празна опашка
- `empty()` — проверка за празнота на опашка
- `enqueue(x)` — включване на елемент в края на опашката
- `dequeue()` — изключване на елемент от началото на опашката
- `head()` — достъп до първия елемент

АТД: опашка

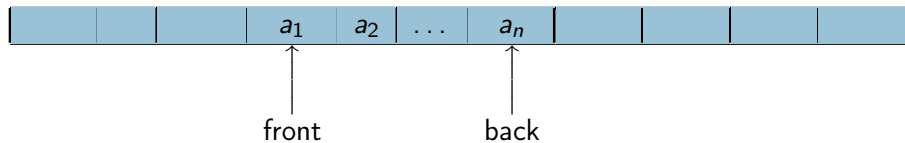
Свойства на операциите

- `create().empty() = true`
- `q.enqueue(x).empty() = false`
- `create().head(), create().dequeue()` — грешка
- `create().enqueue(x1).enqueue(x2)...enqueue(xn).head() = x1`
- `create().enqueue(x1).enqueue(x2)...enqueue(xn).dequeue() = create().enqueue(x2)...enqueue(xn)`

Последователно представяне

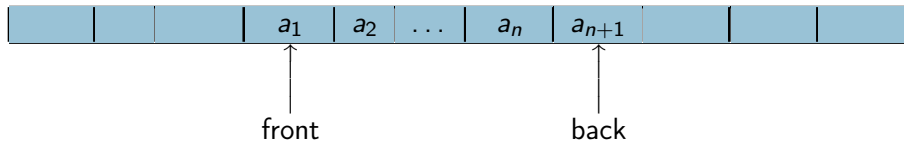


Последователно представяне



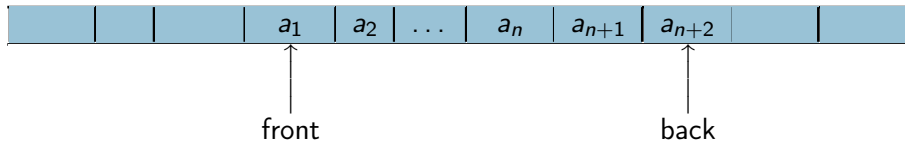
- изключване на елемент (dequeue)

Последователно представяне



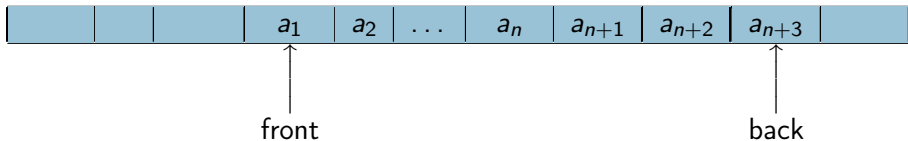
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне



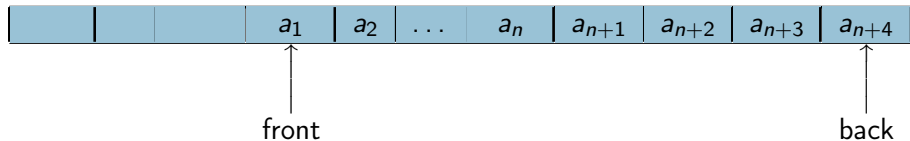
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне



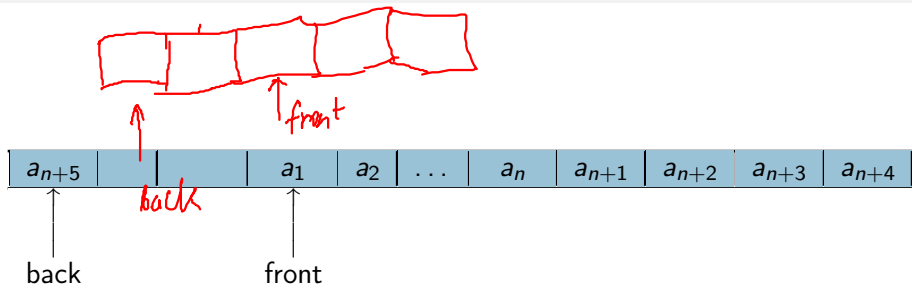
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

Последователно представяне



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

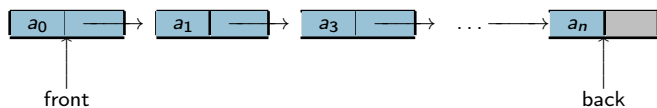
Последователно представяне



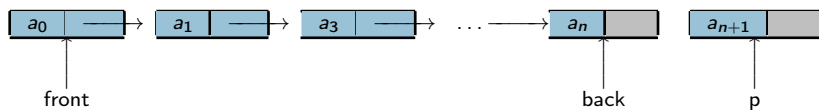
- изключване на елемент (dequeue)
- включване на елемент (enqueue)

;

Свързано представяне

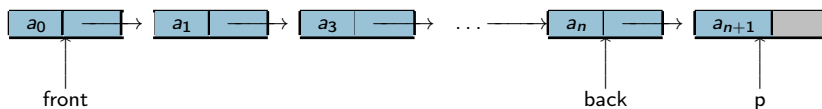


Свързано представяне



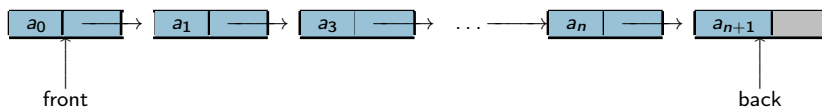
- включване на елемент (enqueue)

Свързано представяне



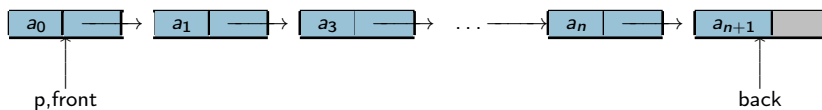
- включване на елемент (enqueue)

Свързано представяне



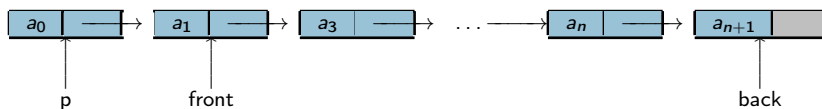
- включване на елемент (enqueue)

Свързано представяне



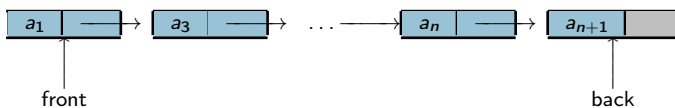
- включване на елемент (enqueue)
- изключване на елемент (dequeue)

Свързано представяне



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

Свързано представяне



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

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.

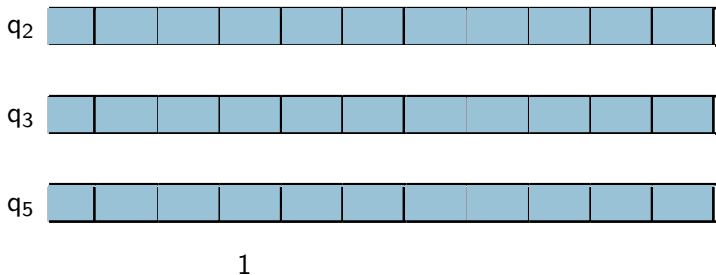
Числа на Hamming

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:



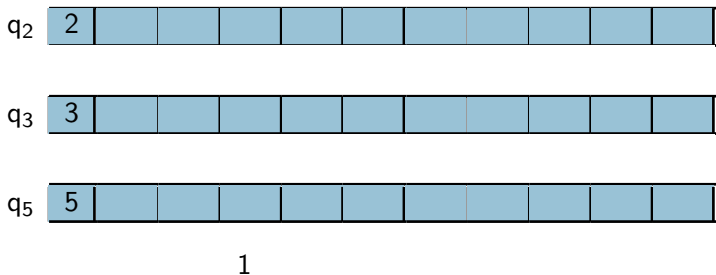
Числа на Hamming

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:



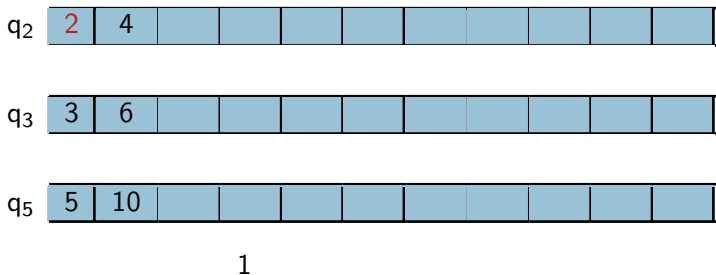
Числа на Hamming

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:



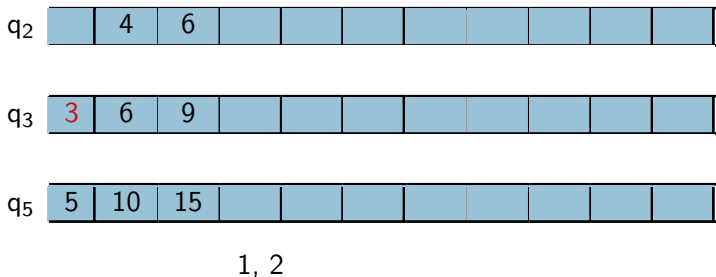
Числа на Hamming

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:



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:

q_2		4	6	8						
-------	--	---	---	---	--	--	--	--	--	--

q_3		6	9	12						
-------	--	---	---	----	--	--	--	--	--	--

q_5	5	10	15	20						
-------	---	----	----	----	--	--	--	--	--	--

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:

q_2			6	8	10					
-------	--	--	---	---	----	--	--	--	--	--

q_3		6	9	12	15					
-------	--	---	---	----	----	--	--	--	--	--

q_5	5	10	15	20	25					
-------	---	----	----	----	----	--	--	--	--	--

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:

q_2			6	8	10	12				
-------	--	--	---	---	----	----	--	--	--	--

q_3		6	9	12	15	18				
-------	--	---	---	----	----	----	--	--	--	--

q_5		10	15	20	25	30				
-------	--	----	----	----	----	----	--	--	--	--

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:

q_2				8	10	12					
-------	--	--	--	---	----	----	--	--	--	--	--

q_3			9	12	15	18					
-------	--	--	---	----	----	----	--	--	--	--	--

q_5		10	15	20	25	30					
-------	--	----	----	----	----	----	--	--	--	--	--

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:

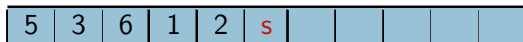


5 1

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:



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).

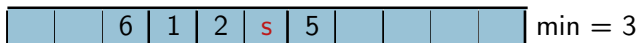
Решение:



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).

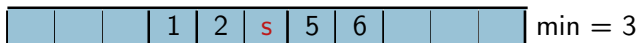
Решение:



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).

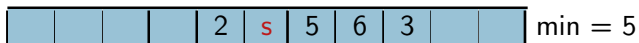
Решение:



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).

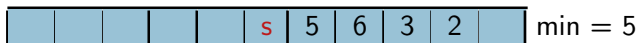
Решение:



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).

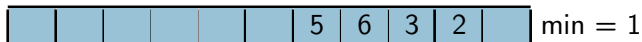
Решение:



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:



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

		⊗			⊗				
		⊗			⊗				⊗
				⊗				⊗	
			⊗		1		⊗		
⊗			⊗			⊗			⊗
⊗	⊗		⊗	⊗		⊗	⊗		
		⊗				⊗			⊗
						⊗			
			⊗					⊗	

Метод на вълната

		⊗			⊗				
		⊗			⊗				⊗
				⊗	2			⊗	
			⊗	2	1	2	⊗		
⊗			⊗		2	⊗			⊗
⊗	⊗		⊗	⊗		⊗	⊗		
		⊗				⊗			⊗
						⊗			
			⊗					⊗	

Метод на вълната

		⊗			⊗				
		⊗			⊗				⊗
				⊗	2	3		⊗	
			⊗	2	1	2	⊗		
⊗			⊗	3	2	⊗			⊗
⊗	⊗		⊗	⊗	3	⊗	⊗		
		⊗				⊗			⊗
						⊗			
			⊗					⊗	

Метод на вълната

		⊗			⊗				
		⊗			⊗	4			⊗
				⊗	2	3	4	⊗	
			⊗	2	1	2	⊗		
⊗			⊗	3	2	⊗			⊗
⊗	⊗		⊗	⊗	3	⊗	⊗		
		⊗			4	⊗			⊗
						⊗			
			⊗					⊗	

Метод на вълната

		⊗			⊗	5			
		⊗			⊗	4	5		⊗
				⊗	2	3	4	⊗	
			⊗	2	1	2	⊗		
⊗			⊗	3	2	⊗			⊗
⊗	⊗		⊗	⊗	3	⊗	⊗		
		⊗		5	4	⊗			⊗
					5	⊗			
			⊗					⊗	

Метод на вълната

						6			
		⊗			⊗	5	6		
		⊗			⊗	4	5	6	⊗
				⊗	2	3	4	⊗	
			⊗	2	1	2	⊗		
⊗			⊗	3	2	⊗			⊗
⊗	⊗		⊗	⊗	3	⊗	⊗		
		⊗	6	5	4	⊗			⊗
				6	5	⊗			
			⊗		6			⊗	

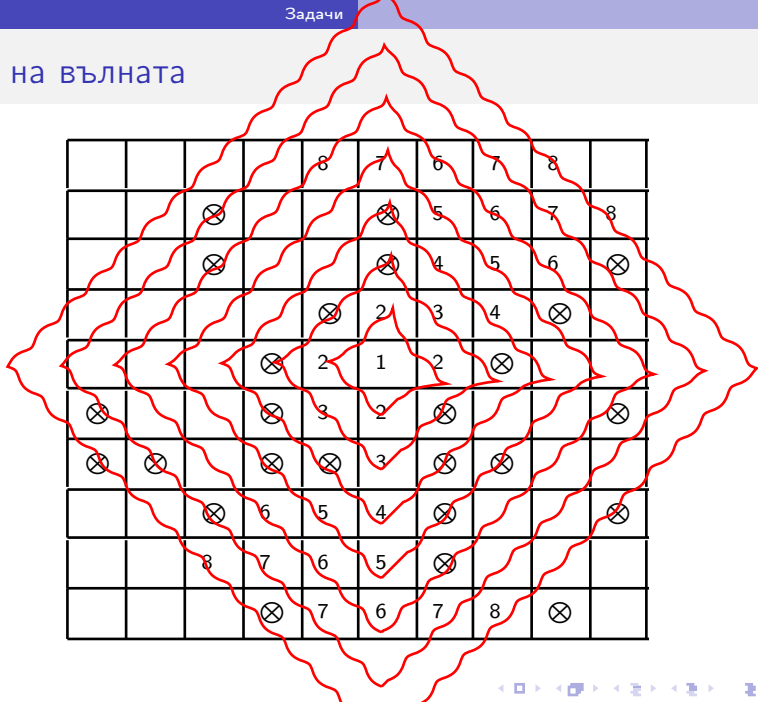
Метод на вълната

					7	6	7		
		⊗			⊗	5	6	7	
		⊗			⊗	4	5	6	⊗
				⊗	2	3	4	⊗	
			⊗	2	1	2	⊗		
⊗			⊗	3	2	⊗			⊗
⊗	⊗		⊗	⊗	3	⊗	⊗		
		⊗	6	5	4	⊗			⊗
			7	6	5	⊗			
			⊗	7	6	7		⊗	

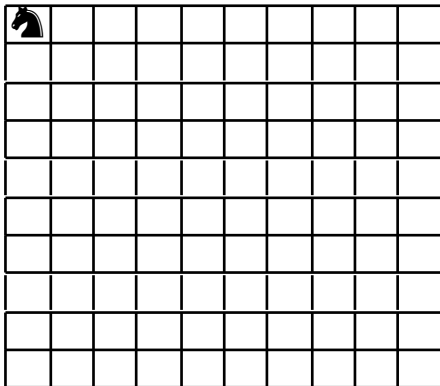
Метод на вълната

				8	7	6	7	8	
		⊗			⊗	5	6	7	8
		⊗			⊗	4	5	6	⊗
				⊗	2	3	4	⊗	
			⊗	2	1	2	⊗		
⊗			⊗	3	2	⊗			⊗
⊗	⊗		⊗	⊗	3	⊗	⊗		
		⊗	6	5	4	⊗			⊗
		8	7	6	5	⊗			
			⊗	7	6	7	8	⊗	

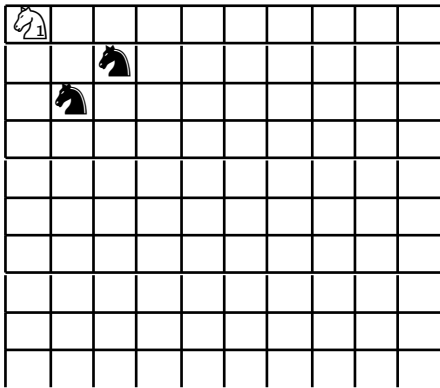
Метод на вълната



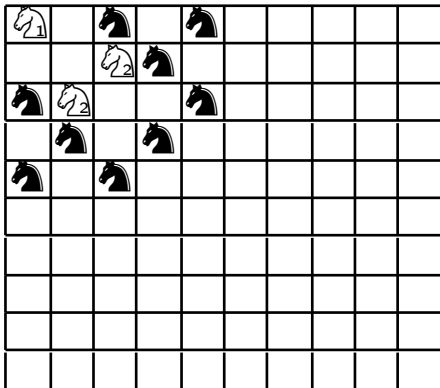
Ход на коня — най-кратък път



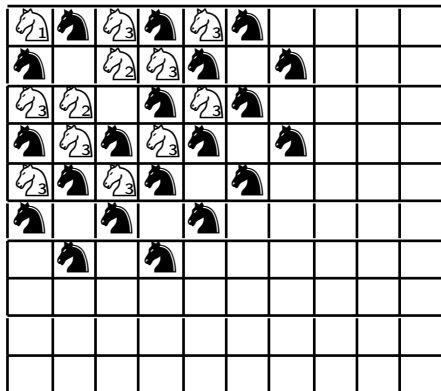
Ход на коня — най-кратък път



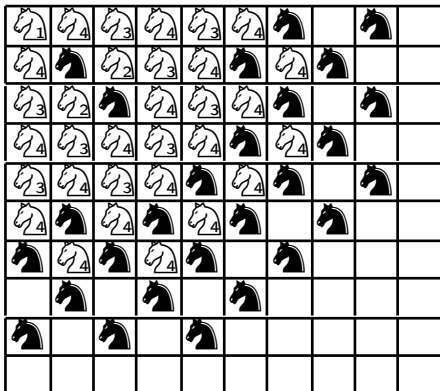
Ход на коня — най-кратък път



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