

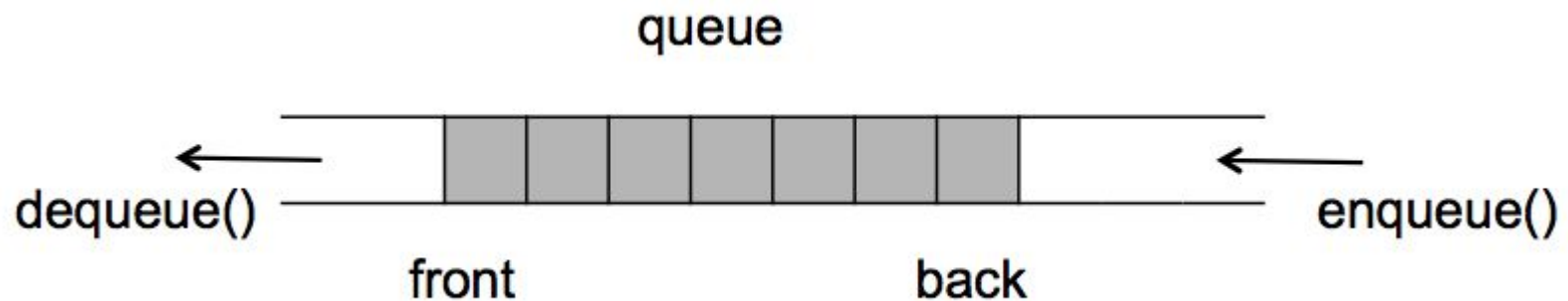
# Черга (Queue)

- Перший прийшов – перший пішов, (FIFO — first in, first out)
- Немає талончиків (див. priority queue, черга з пріоритетами)
- Не можна “просто спитати” (див. SJN, short job next)



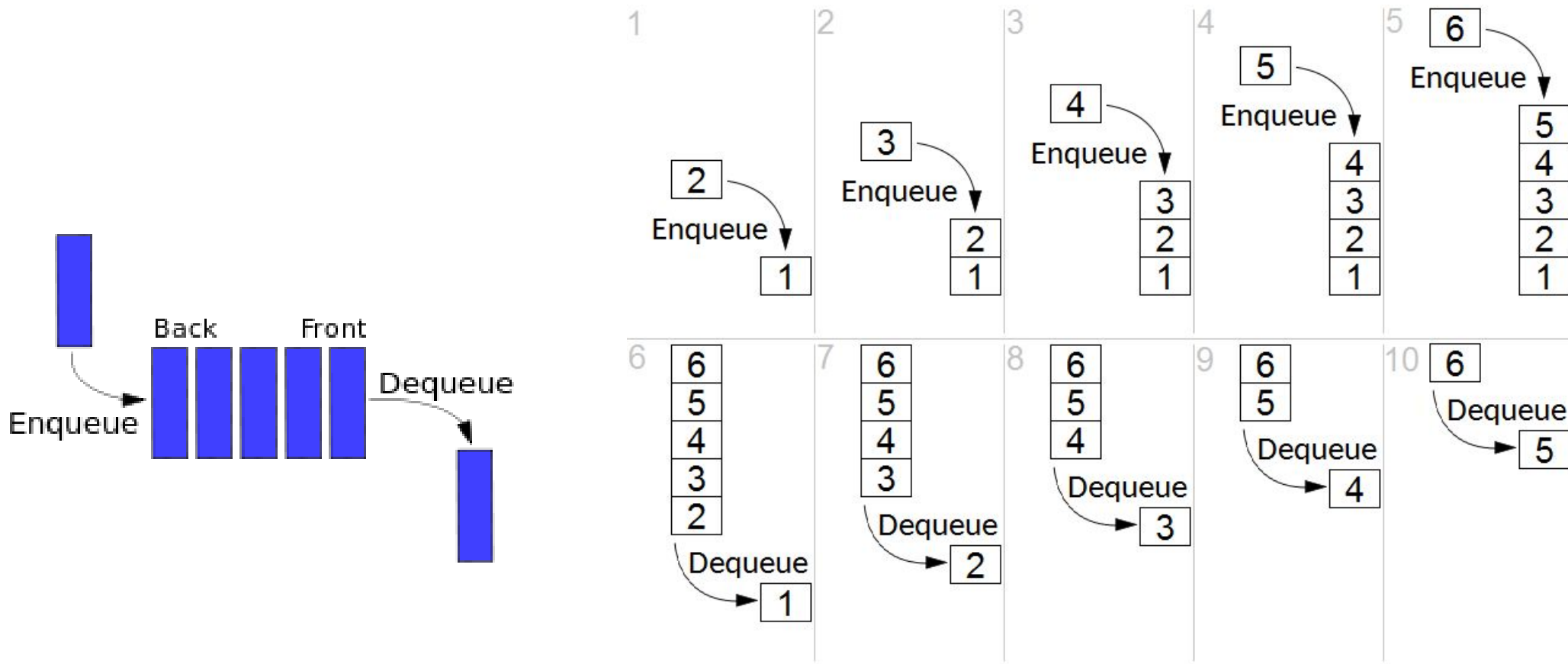
# Черга

- У черги є голова (head) та хвіст (tail).
- Елемент, що додається до черги, опиняється в її хвості.
- Елемент, що видаляється з черги, знаходиться в її голові.

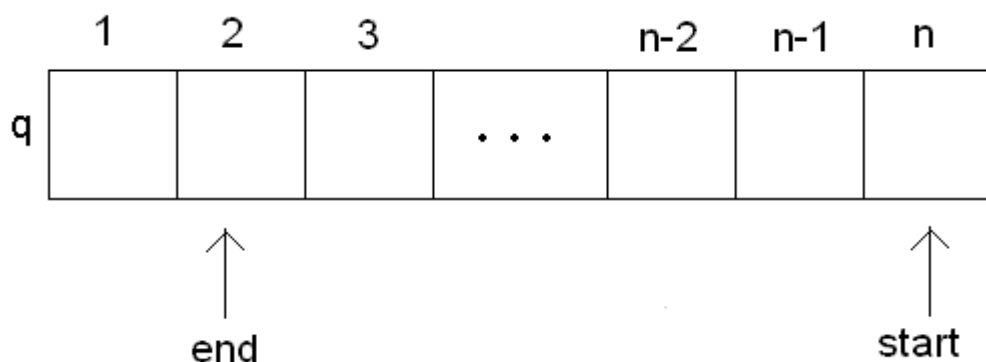


# Основні методи

- **enqueue** — "поставити в чергу". Операція додавання елемента в "хвіст" черги. При цьому довжина черги збільшується на одиницю.
- **dequeue** — "отримання з черги". Операція, яка повертає елемент з голови та видаляє його з черги, таким чином встановлюючи голову на наступний за видаленим елемент та зменшуючи довжину на одиницю.



# Реалізація з масивами (кільцевий буфер)



- **Start** – голова черги, **end** – хвіст
- **Enqueue** – елемент додається у  $q[end]$ ,  $end--$
- Якщо  $end$  стає менше 1, то йому присвоюється значення  $n$ , поки значення  $end$  не досягне  $start$
- **Dequeue** – повертається елемент з  $q[start]$ ,  $start--$

# Реалізація зі зв'язними списками

- Використовується динамічна пам'ять
- Розмір черги не обмежується розміром масиву
- Але на відміну від масиву пам'ять фрагментується і черга працює повільніше



# Реалізація з двома стеками

- В один стек записується, а з другого зчитується
- **Enqueue** – додає елемент в S1
- **Dequeue** – послідовно дістає всі елементи з S1 та перекладає їх в S2, потім повертає верхній елемент з S2

