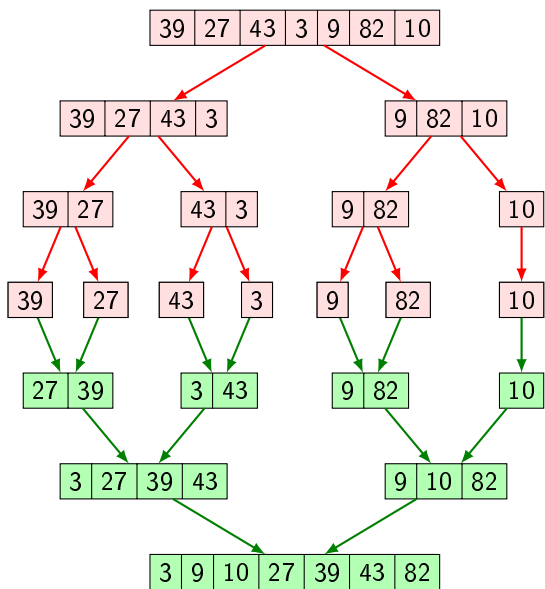


# Templates. Move semantics

## 1 Задача първа



Ще разгледаме алгоритъма за сортиране merge sort. Не е задължително да сортираме числа, нашата задача е да можем да сортираме всеки тип, който поддържа *operator<*.

Merge sort приема масив от обекти. Ако дължината на масива е 1, то не правим нищо, масив с един елемент вече е сортиран. Ако имаме масив с дължина  $n$ , merge sort разбива масива на два подмасива  $Array_{left}[1..\frac{n}{2}]$  и  $Array_{right}[\frac{n}{2} + 1..n]$ . Върху всяка от двете части алгоритъмът рекурсивно извиква себе си. Понеже merge sort е сортиращ алгоритъм, очакваме лявата и дясната част да са сортирани. Сега разполагаме с масив, чийто лява и дясна част са сортирани. Върху тези две части извикваме функцията merge. Тя приема два подмасива, като важното свойство, без което merge става безполезна, е, че

двата подмасива са сортирани. Merge обединява двата сортирани подмасива в нов масив, като използва допълнителна памет, след което копира данните обратно в първоначалния масив. Така разполагаме със сортирана колекция. Примерен псевдокод на merge sort:

```
void mergeSort (Array [1..n]) {  
    if (n == 1)  
        return ;  
  
    mergeSort (Array [1..n/2]);  
    mergeSort (Array [n/2 + 1..n]);  
    merge (Array [1..n/2], Array [n/2 + 1, n]);  
}
```

## 2 Задача втора

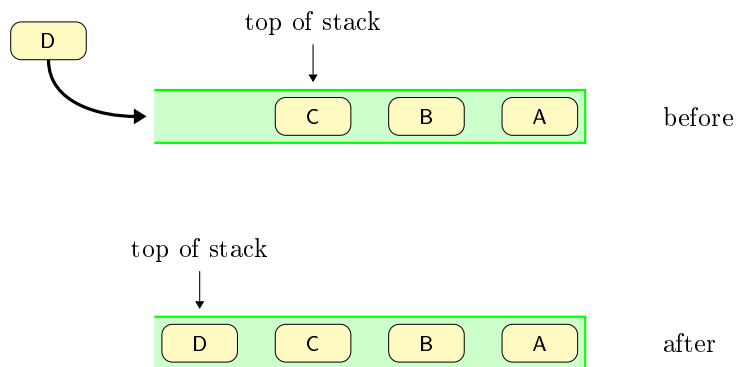
Ще реализираме структурата от данни Stack (Стек). Стекът е колекция от елементи, в която имаме дефинирани следните операции:

1. push(const T&) - добавя елемент към колекцията.
2. pop() - премахва последния добавен елемент от колекцията
3. top() - връща последния добавен елемент.
4. empty() - връща дали колекцията е празна.

Заедно с тази функционалност да се реализират и move семантики за нашия стек.

Важно свойство на стека е, че имаме достъп само до последния добавен елемент в колекцията, още наричан връх на стека (top of stack). Този достъп трябва да бъде в константно време. Имаме ли достъп до други елементи от стека, това вече не е стек.

**Пример за push:**



**Пример за pop:**

