

1. Проиллюстрируйте работу процедуры `HeapExtractMin()` в очереди с приоритетами реализованной с помощью пирамиды, хранящейся в массиве $A = 5, 7, 9, 8, 11, 10, 17, 13$.

 test.jpg X

2. Проиллюстрируйте результат воздействия на изначально пустую очередь Q , хранящуюся в массиве $Q[1..6]$, операций `Enqueue(Q, 4)`, `Enqueue(Q, 1)`, `Enqueue(Q, 7)`, `Dequeue(Q)`, `Dequeue(Q)`, `Enqueue(Q, 5)`. Требуется нарисовать состояние очереди после выполнения каждой из операций, а также отметить положение $head[Q]$ и $tail[Q]$.

Ответ в файле.

3. Разработайте нерекурсивную процедуру со временем работы $\Theta(n)$, обращающую порядок расположения элементов в однократно связанном списке. Процедура должна использовать некоторый постоянный объем памяти помимо памяти, необходимой для хранения всего списка.

Давайте пройдем за $O(n)$ по списку и на каждом шаге будем класть все элементы в `stack<T> a`. Затем пойдем с начала списка и будем заменять значение в узле на `a.top()` (конец "a") и выполняя `a.pop()` (удаление последнего элемента a). Тоже за $O(n)$. Тем самым получим "развернутый" list.

4. Определите асимптотическое время выполнения перечисленных в приведенной таблице операций над элементами динамических множеств в наихудшем случае, если эти операции выполняются со списками перечисленных ниже типов. Если список отсортирован, после выполнения операций он должен оставаться отсортированным.

	Неотсортированный однократно связан- ный список	Отсортированный однократно связан- ный список	Неотсортированный дважды связанный список	Отсортированный дважды связанный список
$Search(L, k)$				
$Insert(L, x)$				

Search:

1. $O(n)$
2. $O(n)$
3. $O(n)$
4. $O(n)$

Insert:

1. $O(1)$
 2. $O(n)$
 3. $O(1)$
 4. $O(n)$
-

5. Покажите, как реализовать очередь с помощью двух стеков.

Проанализируйте время работы операций, которые выполняются с ее элементами.

Пусть у нас есть $stack<T> a$, в котором будут наши элементы давайте засунем все элементы "a" в $stack<T> b \Rightarrow$ мы получили развернутый стек \Rightarrow первый элемент в "a" стал последним \Rightarrow мы получили что-то похожее на очередь.

Давайте удалим верхний элемент в "b" и положим все элементы обратно в "a". Таким образом мы удалили только первый элемент из стека "a".

Удаление за $O(n)$, вставка за $O(1)$.
