1. (a) Истинное время работы increment в худшем случае составит k операций
Это значение достигается когда массив a заполнен единицами.

(b) Посмотрим сколько раз мы затрагивали бит с номером i.
Первый бит меняется каждую операцию, второй бит - каждую вторую. Нетрудно заметить что i бит меняет значение только после 2^{i-1} операций increment. Нулевой бит инвертируется при каждом вызове increment
Итого получаем $n + \frac{n}{2} + \frac{n}{4} + ... + 1 < 2n$

(c) Рассмотрим массив в котором на $k - 1$ месте стоит единица, а остальные значения - нули.
Будем чередуя выполнять операции decrement и increment.
Нетрудно заметить что каждая будет выполняться за k простых операций.
Следовательно время работы в худшем случае — $\mathcal{O}(nk)$.

(d) Заведём дополнительную переменную right, изначально равную -1, — самая правая единичка. После неё в массиве может храниться что угодно, но мы будем считать что там лежат нули.

   i. Когда цикл в increment дойдет до right $+1$ мы должны остановиться так как помним о том что все элементы после right равны нулю.
   Ещё нужно подвинуть right в случае когда $i >$ right

   ii. get(i) нужно изменить так что-бы при $i > operatornameright$ она возвращала 0.

   iii. Теперь setZero будет просто ствить right в значение -1

```
increment():
  i = 0
  while i < k and i <= right and a[i] = 1:
    a[i] = 0
    i++
  right = max(right, i)
  if i < k:
    a[i] = 1

get(i):
  if i > r:
    return 0;
  return a[i]

setZero():
  r = -1
```

2. Придумаем функцию потенциала:
   n — количество элементов в векторе
   c — фактический размер вектора
   $\frac{c}{4} \leq n \leq c$
   $\Phi(n, c) = \begin{cases} c \leq 2n, 2n - c, \\ 2n < c, 2c - n. \end{cases}$
   $\text{push} = \begin{cases} c \leq 2n,\ 1 + \Phi(n+1, c) - \Phi(n, c) = \mathcal{O}(1), \\ 2n < c,\ 1 + \Phi(n+1, c) - \Phi(n, c) = \mathcal{O}(1), \\ n = c,\ n + \Phi(n+1, 2c) - \Phi(n, c) = \\ = n + 2n + 2 - 2c - 2n + c = 3 + n - c = \mathcal{O}(1). \end{cases}$
   $\text{pop} = \begin{cases} c \leq 2n,\ 1 + \Phi(n+1, c) - \Phi(n, c) = \mathcal{O}(1), \\ 2n < c,\ 1 + \Phi(n+1, c) - \Phi(n, c) = \mathcal{O}(1), \\ n = \frac{c}{4},\ c + \Phi(n-1, \frac{c}{2}) - \Phi(n, c) = \\ = c + c - n + 1 - 2c + n = 1 = \mathcal{O}(1). \end{cases} \Rightarrow$
   $\Rightarrow$ все операции выполняются в среднем за $\mathcal{O}(1)$

3.

4. Будем внутри нашей памяти хранить односвязный список не занятых блоков. Заведём указатель head на последний элемент списка. В вершине списка будем хранить одно число — указатель на предыдущий элемент, или -1, если элемент является последним.
   Пусть для нас память представляет собой массив a длины n.
   Проинициализируем её так:
   $\forall i \in \{0, 1, \ldots, n-1\},\ a_i = i - 1$
   $\text{head} = n - 1$

5.