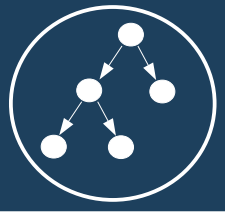


Data Structures and Algorithms

Алгоритмы.
Генерация сочетаний в лексикографическом порядке.

Сведение о алгоритме

Сложность по времени в наихудшем случае $O\left(\frac{n!}{k!(n-k)!}\cdot\frac{n+1}{n+1-k}\right)$

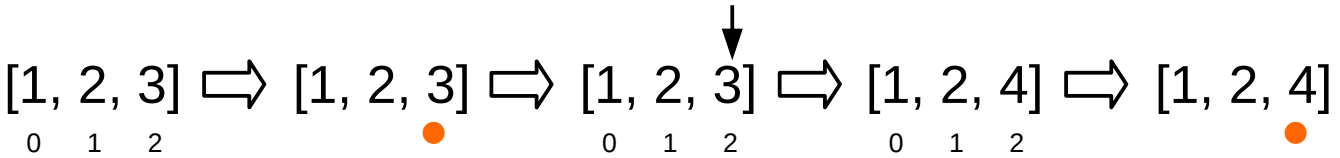
Затраты памяти $O(k)$

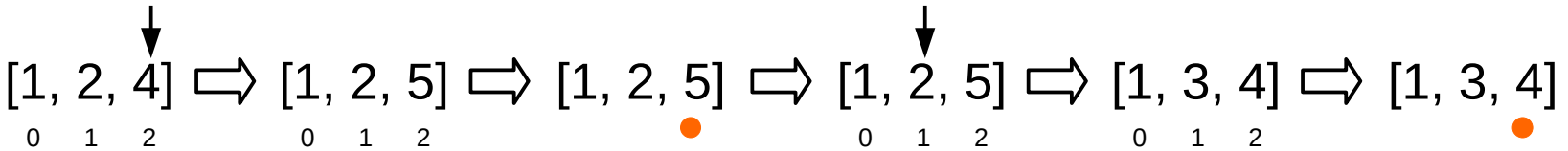
k — элементов выбранных из множества из n - элементов

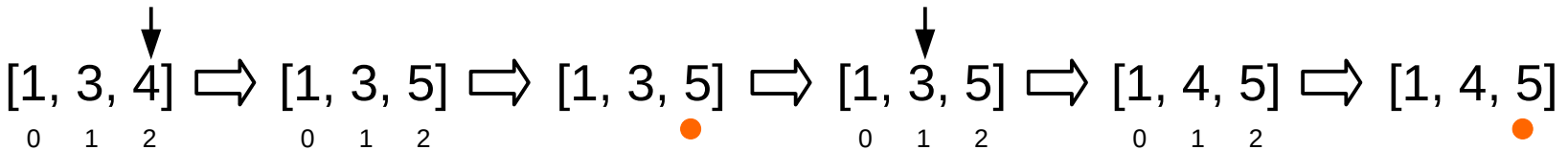
Принцип работы алгоритма

1) Создаем последовательность (в дальнейшем a) размером k элементов. Индексация в последовательности начинается с нуля. Заполняем последовательность значениями от 1 до k. Переходим к пункту **2**.
2) Используем последовательность как очередное сочетание. Переходим к пункту **3**.
3) Начиная с конца последовательности ищем такой элемент, что выполняется условие $a[i] \leq n-k+i$. Если такой элемент найден то переходим у пункту **4**. В противном случае **заканчиваем алгоритм**.
4) Увеличиваем найденный элемент на единицу. От найденного элемента и до конца последовательности устанавливаем значения на единицу больше предыдущего элемента. Переходим к пункту **2**.

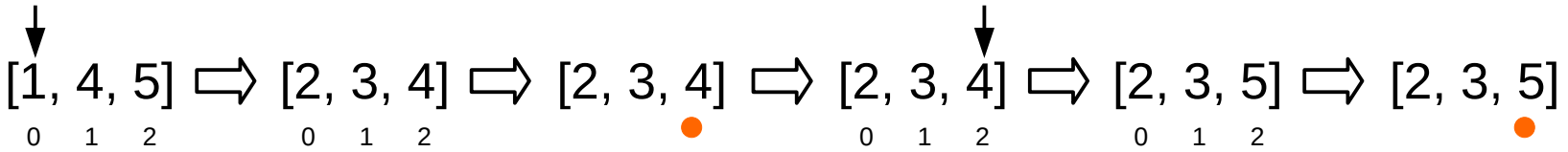
Графическая иллюстрация работы алгоритма (сочетание 3 из 5)

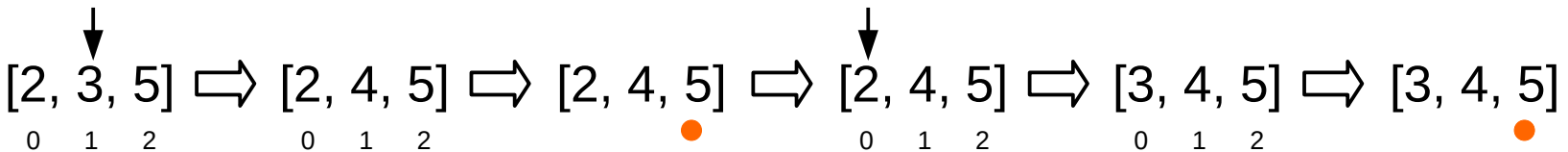


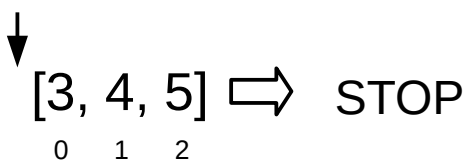




Графическая иллюстрация работы алгоритма (сочетание 3 из 5)







Графические обозначения

- найденный элемент

- готовое сочетание

Python

Реализация алгоритма на Python

Реализация алгоритма на Python

```
def print_combination(k, n):
    comb = [i+1 for i in range(k)]
    while True:
        print(comb)
        m = -1
        for i in range(k-1,-1,-1):
            if comb[i] <= n-k+i:
                m = i
                comb[i]+=1
                break
        if m == -1:
            break
        for i in range(m+1,k):
            comb[i]=comb[i-1]+1
```

Java

Реализация алгоритма на Java

Реализация алгоритма на Java

```
public static void printCombination(int k, int n) {
     int[] comb = new int[k];
     for (int i = 0; i < comb.length; i++) {
          comb[i] = i + 1;
     }
     for (;;) {
          System.out.println(Arrays.toString(comb));
          int m = -1;
          for (int i = k - 1; i >= 0; i--) {
               if (comb[i] <= n - k + i) {
                    comb[i] += 1;
                    m = i;
                    break;
               }
          }
          if (m == -1) {
               break;
          }
          for (int i = m + 1; i < k; i++) {
               comb[i] = comb[i - 1] + 1;
          }
     }
}
```

Fortran

Реализация алгоритма на Fortran

Процедура для генерации сочетаний

```
subroutine print_combination(k,n)
   implicit none
   integer,intent(in)::k,n
   integer::comb(1:k)
   integer::i,m
   comb = [(i,i=1,k)]
   do
      write(*,*) comb
      m = -1
      do i = k,1,-1
         if(comb(i)<n-k+i) then
            comb(i) = comb(i) + 1
            m = i
            exit
         end if
      end do
      if (m==-1) then
         exit
      end if
      do i = m+1,k
         comb(i) = comb(i-1) + 1
      end do
   end do
end subroutine print_combination
```

Список литературы

1) Д. Кнут. Искусство программирования. Том 4. Генерация всех сочетаний и разбиений, 3-е изд.