

ЗАДАНИЕ ПО ИНФОРМАТИКЕ

ВАРИАНТ 73111 для 11 класса

Для заданий 1-3, 5 требуется разработать алгоритм на языке блок-схем, псевдокоде или естественном языке

Век стоит Русь не шатается и века простоит не пошатнётся. Да приключилась новая напасть с землёй Русской. Из заморских путешествий Колыван вернулся, и затеяли они с Бабой-Ягой дело тёмное.

1. Пошли бояре по приказу Князя в библиотеку искать информацию о немецких дворянских титулах, а там Моисей систему поиска информации по четырём знакам придумал. Каждый знак состоит из нескольких одинаковых черт, которые располагаются по сторонам прямоугольного шаблона (см. рисунок). Да только ограничил его в финансах Юлий и нет возможности поставить полноценную систему анализа знаков. На каждую черту требуется один датчик, показывающий прочерчена или не прочерчена черта. Достать бояре обещали датчики, но только минимальное количество, необходимое для работы системы поиска информации, придуманной Моисеем. Сколько датчиков необходимо добыть боярам для работы системы?



Решение. Для распознавания четырёх объектов, по идее, достаточно рассматривать две черты, т.к. будут существовать четыре комбинации прочерчена/не прочерчена. Но необходимо выбрать черты, которые различны в разных знаках. Для данной комбинации знаков не существует такой пары, которая давала бы четыре различные комбинации. Поэтому придётся использовать три черты. Левую и правую использовать не будем, т.к. они одинаковы во всех знаках. Из оставшихся можно взять две нижних и одну (любую) из верхних.

2. Завлёл Колыван Князя в спор, по наущению Яги, да и загадал задачку сложную: «В ячейках колонки длины N (N ячеек) написаны целые числа. Найти k -ое по возрастанию число среди чисел, записанных в этих ячейках. Обеспечить число операций порядка N^2 ». Князь так увлёкся задачкой, что не заметил, как подписал указ о передаче княжеского дворца Яге и Колывану. Помогите Князю решить задачку, чтобы снова вернуться к делам государственной важности.

Решение – 1 способ. Используем быструю сортировку для сортировки массива и берём k -ое число из отсортированного массива.

```
алг KMin
нач
  цел n, k, i
  цел x[n]

  ввод n
  если n <= 0 то
    вывод 'Неверное значение n'
  иначе
    для i от 1 до n
      нц
        ввод x[i]
      кц
    ввод k
    QuickSort(x, 1, n)
    вывод x[k]
  всё
кон
```

```
алг QuickSort(арг рез цел x[n], арг цел n1, арг цел n2)
нач
  цел i, j, k, y

  k = x[(n1 + n2) div 2]
  i = n1
  j = n2
```

Олимпиада школьников «Надежда энергетики». Заключительный этап. Очная форма.

```
повторять
пока x[i] < k
нц
  i = i + 1
кц
пока x[j] > k
нц
  j = j - 1
кц
если i < j то
  y = x[i]
  x[i] = x[j]
  x[j] = y
  i = i + 1
  j = j - 1
иначе
  если i = j то
    i = i + 1
    j = j - 1
  всё
всё
до i > j
если n1 < j то
  QuickSort(x, n1, j)
если i < n2 то
  QuickSort(x, i, n2)
кон
```

Решение – 2 способ. Находим минимальный элемент массива. Переставляем его на первое место. Находим минимальный элемент среди элементов от 2-го до последнего. Переставляем его на второе место. Повторяем эту процедуру k раз. k -ый минимальный элемент и будет искомым.

Можно немного усовершенствовать этот метод – искать k -ый минимум, если $k \leq n / 2$, и $(n - k + 1)$ -ый максимум в противном случае.

```
алг KMin
нач
  цел n, k, i, j, min, im
  цел x[n]

  ввод n
  если n <= 0 то
    вывод 'Неверное значение n'
  иначе
    для i от 1 до n
      нц
        ввод x[i]
      кц
    ввод k

    для i от 1 до k
      нц
        min = x[i]
        im = i
        для j от i + 1 до n
          нц
            если x[j] < min то
              min = x[j]
              im = j
          всё
        кц
        y = x[i]
        x[i] = x[im]
        x[im] = y
      кц
    вывод x[k]
  всё
кон
```

В большинстве случаев первый алгоритм будет работать быстрее, чем второй.

3. Решили богатыри срезать путь в Киев через ущелье, да только уже ждали их там враги. Обрезали верёвки и завалили ущелье брёвнами, погребены под брёвнами оказались богатыри и ни выбраться им. С другой стороны в условленном месте ждал Горыныч. Не дождавшись, полетел он искать друзей своих да увидел обвал странный в ущелье и ринулся на разведку. Облетев ущелье,

Олимпиада школьников «Надежда энергетики». Заключительный этап. Очная форма.

выяснил Горыныч, что бревна рядком друг на друга попадали. Одна из голов запомнила координаты расположения начала брёвен и длину ущелья. Другая голова определила, что бревна лежат параллельно друг другу и длина у них одинаковая, да и брёвен n штук. А третья голова, предположив, что богатыри под самым большим скоплением брёвен должны быть (иначе бы они уже выбрались), принялась вычислять, на каком участке ущелья находится самое большое скопление брёвен. Помогите Горынычу написать алгоритм поиска максимальной толщины завала, чтобы он мог спасти друзей своих верных.

Решение. Запишем координаты начала брёвен массив. Добавим в этот массив координаты концов брёвен, вычисляя их по координатам начала и длине бревна lb . Для отличия координаты начала от координаты конца можно использовать отрицательные числа, например, для координат конца (т.е. координата на самом деле положительна, а знак «минус» означает, что это конец бревна). Отсортируем массив так, чтобы модули элементов располагались по неубыванию, но при этом отрицательное число должно стоять раньше такого же по модулю положительного, поскольку левое бревно должно заканчиваться раньше, чем начинается правое. Если голова была разумной, то можно считать, что координаты начала сразу были упорядочены и нужно добавить координаты концов в отсортированный массив.

Положим максимальное количество слоёв k_{\max} и текущее количество слоёв k равными 0. Просматриваем отсортированный массив. Если координата оказывается больше длины ущелья L , можно выйти из цикла, т.к. брёвна за концом ущелья нас не интересуют. Если текущий элемент массива больше или равен 0 (считаем, что самая первая координата может быть нулевой, т.е. бревно может находиться в самом начале ущелья), то увеличиваем количество слоёв k на 1, в противном случае – уменьшаем k на 1. При увеличении k проверяем: если k больше максимального количества слоёв k_{\max} , то меняем k_{\max} и запоминаем текущий элемент массива как координату начала отрезка, покрытого максимальным количеством брёвен.

алг ТолщинаЗавала

```
нач
цел n, lb, L, k, kmax, xmax, i
цел x[2 * n]

ввод n, lb, L
если n <= 0 или lb <= 0 или L <= 0 то
    вывод 'Неверное значение'
иначе
    для i от 1 до n
        нц
            ввод x[i]
            x[i + n] = -(x[i] + lb)
        кц

    InsertSort(x, n)           // Условие в подпрограмме сортировки записано так, чтобы сортировка
производилась                // по возрастанию модулей, а в случае равенства модулей отрицательное число
                                // должно оказаться перед положительным

    kmax = 0
    imax = 0
    k = 0
    i = 1
    пока i <= 2 * n и abs(x[i]) <= L
        нц
            если x[j] >= 0 то
                k = k + 1
                если k > kmax то
                    kmax = k
                    imax = x[j]
            всё
            иначе
                k = k - 1           // x[j] < 0
            всё
            i = i + 1
        кц
    вывод kmax, ' ', imax
всё
кон
```

Олимпиада школьников «Надежда энергетики». Заключительный этап. Очная форма.

алг InsertSort(арг рез цел $x[n]$, арг цел n)

```

нач
  цел  $i, j, y$ 

  для  $i$  от 2 до  $n$ 
  нц
     $y = x[i]$ 
     $j = i - 1$ 
    пока  $j \geq 1$  и ( $\text{abs}(x[j]) > \text{abs}(y)$  или  $\text{abs}(x[j]) = \text{abs}(y)$  и  $x[j] > y$ )
    нц
       $x[j + 1] = x[j]$ 
       $j = j - 1$ 
    кц
     $x[j + 1] = y$ 
  кц
кон
    
```

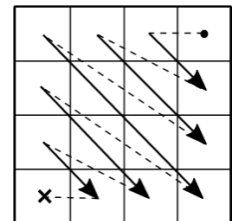
4. Выбрались богатыри из завала и поспешили в Киев, а по пути попался им мост сломанный. Надо чинить его, да только задумались богатыри, сколько древесины потребуется для этого. А Горыныч возьми и вспомни, что: «Существует система из четырёх гирь, таких, чтобы, используя их по одной, можно было взвешивать любой груз Q (целое число кг) в диапазоне от 1 до 40 кг. При этом гири могут быть помещены на обе чаши весов. Первая чаша содержит взвешиваемый груз Q , вторая чаша свободна от груза». Помогите богатырям решить аналогичную задачу для $Q = 120$ кг (укажите минимальное число гирь и их веса).

Решение. Для каждой гири существует три варианта использования – положить гирю на одну чашу весов, положить гирю на другую чашу весов, не использовать гирю. Поэтому для формирования значения, соответствующего весу груза, можно использовать троичную систему счисления. Тогда каждая гиря должна иметь вес, равный значению 3 в некоторой степени, аналогично тому, как в позиционной системе счисления цифра, находящаяся в k -ом разряде (k отсчитывается с 0 справа налево), имеет вес, равный основанию системы счисления в k -ой степени. Таким образом, нужны гири весом $3^0, 3^1, 3^2, 3^3$ и 3^4 , т.е. 1, 3, 9, 27 и 81. Для взвешивания груза весом до 120 кг их будет достаточно. Значения веса будут формироваться следующим образом:

```

1 = 1
2 = 3 - 1 (гири кладутся на разные чаши весов)
3 = 3
4 = 3 + 1
5 = 9 - 3 - 1
6 = 9 - 3
7 = 9 - 3 + 1
8 = 9 - 1
9 = 9
...
39 = 27 + 9 + 3
40 = 27 + 9 + 3 + 1
41 = 81 - 27 - 9 - 3 - 1
...
119 = 81 + 27 + 9 + 3 - 1
120 = 81 + 27 + 9 + 3
    
```

5. Выпустила Баба-Яга Двоих из Ларца Неприветливых с Лица, чтобы Князя и Юлия изловить и в темницу бросить. Бросились убежать они, да забежали во внутренний двор дворца. Помнил Князь, что выход там тайный есть, да вот запомнил где конкретно. А Юлий со страху возьми и вспомни, что в библиотеке в летописях Моисея видел странную запись, ну и озвучил её: «Имеется квадратная таблица размером $N \times N$. В ячейках таблицы записаны целые числа. При этом движение по таблице возможно только в соответствии со схемой на рисунке». Юлий и картинку нарисовал и путь движения указал, и добавил, что координаты последнего положительного элемента при таком пути движения и есть тайный выход, а если идти другим путём, то выход будет заблокирован навечно. Князь с удивлением заметил на земле клеточки и цифры в них и обратился к Юлию: «Так что ты там про таблицу с



Олимпиада школьников «Надежда энергетики». Заключительный этап. Очная форма.

числами говорил?» Помогите Юлию и Князю составить алгоритм поиска номеров строки и столбца нужной ячейки таблицы, чтобы найти тайный выход и спастись от Двоих из Ларца Неприветливых с Лица.

Решение. Поскольку надо найти последний положительный элемент, организуем проход по таблице в обратном порядке с досрочным выходом из цикла. У нас есть $2 \cdot N - 1$ диагоналей, длина которых меняется от 1 до N и затем обратно к 1. Поэтому организуем два цикла по k – от 1 до N и от $N - 1$ до 1. Номер строки i в первых N диагоналях меняется от N до $N - k + 1$. Номер столбца вычисляется как $i + k - N$. В оставшихся диагоналях номер строки i меняется от k до 1. Номер столбца вычисляется как $i + N - k$.

Можно не вычислять номер столбца, а ввести ещё одну переменную j , которая будет начинаться с k в первых циклах и с N – в остальных, и будет уменьшаться на 1 на каждом шаге цикла.

алг Тайный выход

нач

```
цел n
цел table[n , n]
цел k, i, ip, jp
лог found
```

ввод n

если $n \leq 0$ то

 вывод 'Неверное значение n'

иначе

 для i от 1 до n

 нц

 для j от 1 до n

 нц

 ввод table[i, j]

 кц

 кц

found = false

k = 1

пока $k \leq n$ и не found

нц

 i = n

 пока $i \geq n - k + 1$ и не found

 нц

 если $x[i, i + k - n] > 0$ то

 ip = i

 jp = i + k - n

 found = true

 всё

 i = i - 1

 кц

 k = k + 1

кц

k = n - 1

пока $k \geq 1$ и не found

нц

 i = k

 пока $i \geq 1$ и не found

 нц

 если $x[i, i + n - k] > 0$ то

 ip = i

 jp = i + n - k

 found = true

 всё

 i = i - 1

 кц

 k = k - 1

кц

если found то

 вывод ip, jp

иначе

 вывод 'Выхода нет'

всё

кон