

ЗАДАНИЕ ПО ИНФОРМАТИКЕ ВАРИАНТ 73101 для 10 класса

Для заданий 2, 3, 4, 5 требуется разработать алгоритм на языке блок-схем, псевдокоде или естественном языке

1. Утверждения $A \rightarrow C$, $A \& B \rightarrow D$, $\neg B \rightarrow E$ истинны. Чему равны A и B , если C , D и E ложны?

Схема решения. Таблица истинности для логической функции «импликация» представлена ниже.

X	Y	$X \rightarrow Y$
ложь	ложь	Истина
ложь	истина	Истина
истина	ложь	Ложь
истина	истина	Истина

Из таблицы видно, что если следствие ложно, то для того, чтобы вся формула была истинной, необходимо, чтобы посылка также была ложна. Таким образом, утверждение A должно быть ложно, в этом случае $A \& B$ также будет ложно. Кроме того, ложно должно быть $\neg B$, т.е. утверждение B должно быть истинно.

2. В археологических раскопках в Крыму при строительстве трассы «Таврида» археологи

нашли табличку с таким текстом:
$$\sqrt{7} = 2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{4 + \dots}}}}$$

Далее в формуле структура из 1, 1, 1, 4 повторяется бесконечное число раз. Пожалуйста, проверьте записанное предположение – разработайте алгоритм проверки на ЭВМ с точностью до 0.0001 справедливости этой формулы.

Схема решения. Данную дробь можно вычислить по формуле $f_1 = 1 / (1 + 1 / (1 + 1 / (1 + 1 / (4 + f_0))))$, где f_0 – отбрасываемая часть, обозначенная многоточием. Положим сначала f_0 равной 0 и вычислим f_1 . Если разность по модулю между переменными f_0 и f_1 окажется меньше 0.0001, можно прекращать вычисления. Иначе положим переменную f_0 равной f_1 и снова вычислим f_1 по той же формуле. Итоговый результат равен $2 + f_1$.

3. В прямоугольном зале размера $M \times N$ квадратных одинаковых плит движется робот. Изначально робот находится в указанном углу (Н). Робот за 1 ход может передвигаться по одной из четырёх траекторий, показанных на рисунке. В таблице размера $M \times N$ указано, какие плиты заняты колоннами (X). Разработайте алгоритм, отвечающий на вопрос: может ли робот добраться до указанной на рисунке плиты (К). Выход за стены зала, а также проход сквозь колонну не являются возможными.

Траектории движения робота

Схема зала (колонны расставлены условно)

Схема решения. Из каждой точки робот может, в принципе, пойти четырьмя путями. Однако, некоторые пути могут быть заблокированы. Поэтому на каждом шаге надо проверять, может ли робот попасть в каждую из четырёх возможных точек.

Запишем в массив координаты начальной точки. Далее в цикле извлекаем (удаляем) из массива координаты первой находящейся в нём точки, взамен кладём в него координаты точек, куда робот может попасть из этой точки. Таких точек может оказаться от 0 до 4. Если среди них есть конечная точка, значит, робот может туда добраться. Если же робот в принципе не может добраться до конечной точки, то в какой-то момент массив станет пустым, и можно будет прекратить поиск.

4. В поисках яблок шинигами Рюк попал в лабиринт. На стене он увидел изображение схемы лабиринта (квадратная таблица размером $N \times N$). Стрелками на рисунке обозначен путь, по которому можно пройти лабиринт, не попав в ловушку. Если Рюк попадёт в ловушку, то он исчезнет из лабиринта. На каждом шаге при движении по лабиринту Рюку встречаются яблоки с вырезанными на них натуральными числами, которые можно собирать. Помогите шинигами Рюку пройти по лабиринту из правого нижнего угла в левый верхний, подбирая при этом только вкусные яблоки (т.е. те, на которых вырезаны простые числа). Число называется простым, если оно делится только на само себя и на 1.

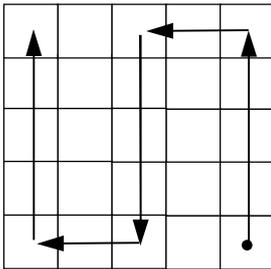


Схема решения. При такой схеме лабиринта проход из правого нижнего угла в левый верхний возможен, только если количество столбцов N выражается формулой $4K + 1$, где K – любое число от 0. Поэтому если $N - 1$ не делится нацело на 4, то проход не возможен. Иначе вычисляем K , это будет количество циклов. В каждом цикле идём из по текущему столбцу j (j начинается с N) снизу-вверх, затем обрабатываем ячейку с индексами $(1, j - 1)$, затем идём по столбцу $j - 2$ сверху вниз, и, наконец, обрабатываем ячейку с индексами $(N, j - 3)$. Столбец с номером 1 проходится снизу вверх после завершения обработки остальных столбцов. При проходе проверяем числа в обрабатываемых ячейках на простоту. Если число является простым, выводим его. Можно сохранять найденные простые числа в массив, а потом вывести этот массив.

Чтобы проверить, что число k является простым, необходимо перебрать его возможные делители от 2 до \sqrt{k} и проверить, делится ли число k на какой-либо из возможных делителей. Если это так, то число не является простым. При этом число 2 является простым, а число 1 не является простым.

Для сокращения перебора можно сначала проверить делимость числа k на 2, а потом проверить его делимость на возможные нечётные делители от 3 до \sqrt{k} с шагом 2. Не забываем, что само число 2 является простым.

Можно ещё больше сократить перебор, отбрасывая числа, которые делятся на 2 и на 3. Для этого для числа k проверяются возможные делители i от 5 до \sqrt{k} с шагом 6, но на каждом шаге цикла проверяется делимость числа k на i и на $(i + 2)$, т.е. получается ряд 5, 7, 11, 13 и т.д. До цикла надо проверить делимость числа n на 2 и на 3, а также надо учесть, что сами числа 2 и 3 являются простыми.

5. В таблице размером $N \times 2$ записаны координаты точек на плоскости (x, y) . N достаточно велико. Все точки лежат на графике некоторой функции, но их порядок нарушен. Таким образом, функция задана табличным способом. Разработайте алгоритм проверки того, что эта функция является монотонно убывающей.

Схема решения. Отсортируем точки по x -координате. Учитывая, что N велико, необходимо использовать эффективные алгоритмы сортировки. После этого проверим, что y -координата каждой точки, начиная со второй, не больше y -координаты предыдущей точки.