

Теоретические задачи для старшей группы.

Указания и краткие решения

1. В компьютере завёлся не очень злобный вирус. Раз в минуту он прерывает нормальную работу компьютера и предлагает пользователю ввести код нейтрализации. Если пользователь введёт правильный код, вирус прекращает свою работу, если код неправильный, то к коду нейтрализации вирус прибавляет или из него вычитает единицу, и вирус продолжает работу. Код нейтрализации – это трёхзначное неизвестное нам число в диапазоне от 000 до 999. Известно, что прибавление и вычитание единицы оставляет число в этом диапазоне. Может ли пользователь гарантированно остановить работу вируса, и если да, то как?

Ответ. Может.

Решение. Предположим сначала, что код нейтрализации – чётный. Вводим 000. Если вирус не остановлен, то код стал нечётным. Вводим 001. Если вирус не остановлен, то код стал чётным, но не 000. Вводим 002, и т.д. до 999. Заметим, что код не может стать меньше последнего введённого нами числа. Значит, при нашем предположении вирус будет остановлен.

Если действие вируса не прекратилось, значит, наше предположение было неверно. Код был нечётным, и после 1000 шагов он опять стал нечётным. В этом случае, надо вводить коды подряд, начиная с 001.

2. При написании программы используются служебные слова (if, then, goto, begin, end, и многие другие). Список всех служебных слов известен. В трансляторе все эти слова должны быстро распознаваться. Предложите способ организации словаря всех служебных слов и способа поиска в нём нужного слова. Предложенные решения будут оцениваться как с точки зрения скорости работы, так и с точки зрения объема используемой памяти. (Все слова образованы малыми латинскими символами, длина каждого не превышает 100, слов может быть сколько угодно.)

Решение.

Понятно, что словарь должен быть отсортирован. Но, если это сделать просто, отсортировав слова в лексикографическом порядке, то поиск в таком словаре осуществляется путём сравнений слов из словаря с искомым словом побуквенно (хотя языки программирования представляют эту процедуру как одну операцию – сравнение слов. Поэтому, простая сортировка слов не даёт самого эффективного решения задачи. Воспользуемся структурой данных – бор, который представляет собой корневое дерево. В каждой вершине дерева будем хранить массив из 26 указателей на потомков, каждый элемент которого соответствует переходу по букве от 'a' до 'z'. Если потомка соответствующего данной букве нет, то храним пустую ссылку, то есть у каждой вершины дерева будет не более 26 рёбер в потомков. Слово хранится как путь от корня дерева до некоторой вершины (если сложить буквы на данном пути в порядке от корня). Для корректного нахождения слов в каждой вершине необходимо хранить флаг, показывающий является ли данная вершина концом слова или нет. В такой структуре поиск осуществляется значительно эффективнее.

Теоретические задачи для старшей группы.

Указания и краткие решения

3. Для тестирования некоторой программы требуется сгенерировать двумерный числовой массив (таблицу) размера $N \times N$, в котором встречаются все натуральные числа от 1 до N^2 в произвольном порядке. Изначально в массиве стоят нули. Для генерации теста разрешено использовать только одну операцию: выбрать два клетки соответствующей таблицы, отстоящие друг от друга на ход коня, и к содержимым этих клеток прибавить по единице. Опишите алгоритм, как это сделать, если это возможно. Размер таблицы задаётся.

Решение.

Заметим, что при нечётном N решения у задачи нет, т.к. в этом случае сумма всех чисел в таблице должна стать нечётной, а прибавляя 2 единицы к содержимому клеток, мы оставляем эту сумму чётной.

В случае чётного N достаточно разбить все клетки таблицы на четвёрки, соответствующие трём ходам коня. Это сделать легко. При этом необязательно проходить конём все клетки доски по одному разу. Пусть A-B-C-D – одна из таких четвёрок. Сначала числа в этой цепочке нулевые, т.е. 0-0-0-0.

Прибавляя единицы по правилам, последовательно получим: 1-1-0-0, 1-2-1-0, 1-2-2-1, 1-2-3-2, 1-2-4-3. Итак, в четвёрке можно получить 4 последовательных числа. Если надо начинать не с нуля, а с какого-то K , то дополнительно к парам A-B и C-D надо $K-1$ раз добавить по единице.