Разбор задач Первой Интернет-олимпиады

Автор: Федор Царев

Введение

В базовой номинации Первой Интернет-олимпиады сезона 2006-2007 участникам было предложено для решения 9 задач. В олимпиаде приняло участие 135 команд, из них 123 решили хотя бы одну задачу.

Наиболее простой оказалась задача «E. Число e» — ее решили 118 команд. Наиболее сложными — задачи «F. Карта» (ее решили 26 команд) и «H. Гонки» (ее решили 19 команд).

Условия задач, результаты олимпиады, тесты и решения жюри можно найти на сайте интернет-олимпиад `http://neerc.ifmo.ru/school/io`.

Задача A. Алхимия

Для решения этой задачи построим ориентированный граф, вершинами которого будут химические вещества. Два вещества будут соединены дугой, если из первого можно получить второе за одну алхимическую реакцию.

Исходную задачу теперь можно сформулировать следующим образом: «Найти кратчайший путь между двумя заданными вершинами в построенном графе». Так как граф не является взвешенным, то для решения этой задачи применяется поиск в ширину [2].

Некоторую трудность может вызвать построение графа. Необходимо учитывать, что исходное и требуемое вещество могут быть не упомянуты в заданных алхимических реакциях.

Задача B. Скобочки

Для начала приведем алгоритм проверки того, является ли некоторая строка $S = s_1s_2\ldots s_n$ правильной скобочной последовательностью.

Для этого воспользуемся стеком [3]. Будем последовательно просматривать символы строки слева направо и их обрабатывать: если встретилась открывающая скобка, то ее необходимо добавить в стек. Если же встретилась закрывающая скобка, то необходимо проверить, что на верхушке стека «лежит» соответствующая ей открывающая — если это так, то открывающая скобка удаляется из стека (в противном случае последовательность не является правильной скобочной). Кроме этого, после обработки всей строки необходимо проверить, что стек пуст (в противном случае последовательность не является правильной скобочной).

После этого задача в силу не очень большой длины заданной строки сводится к перебору всех циклических сдвигов строки (их количество равно длине строки) и применению к каждому из них описанного выше алгоритма.

Задача C. Наилучший делитель

Решение этой задачи состоит в простом переборе всех чисел от 1 до n, проверки, являются ли они делителями n и подсчета суммы цифр.

Ниже приведен фрагмент программы на языке Pascal, реализующий этот подход.

```pascal
ans := 1;
bestSum := 1;
for i := 1 to n do begin
  if n mod i = 0 then begin
    sum := 0;
    x := i;
    while x > 0 do begin
```

```
      sum := sum + (x mod 10);
      x := x div 10;
    end;
    if (sum > bestSum) or ((sum = bestSum) and (i < ans)) then begin
      ans := i;
      bestSum := sum;
    end;
  end;
end;
```

Задача D. Наихудший делитель

Заметим, что у любого числа существует делитель с суммой цифр, равной единице, — это сама единица. Таким образом, у числа-претендента на то, чтобы быть наихудшим делителем, сумма цифр должна быть равна единице. Кроме единицы, таким свойством обладают степени десяти (числа вида 10^k, $k \geq 1$).

Таким образом, задача свелась к нахождению максимальной степени десяти, на которую делится заданное число n. Так как n задано в десятичной системе счисления, то эта задача эквивалентна задаче нахождения числа нулей, на которое заканчивается запись числа n.

Задача E. Число e

Решение задачи состоит в непосредственной реализации процесса округления числа.

Этот подход реализует приведенный ниже фрагмент программы.

```
read(n);
e = '7182818284590452353602875O';
if (n = 0) then begin
  writeln(3)
end else begin
  write('2.');
  c := 0;
  if (ord(e[n + 1]) - ord('0') >= 5) then begin
    a[n] := (ord(e[n]) - ord('0') + 1) mod 10;
    c := (ord(e[n]) - ord('0') + 1) div 10;
  end else begin
    a[n] := ord(e[n]) - ord('0');
  end;
  for i := n - 1 downto 1 do
  begin
    a[i] := (ord(e[i]) - ord('0') + c) mod 10;
    c := (ord(e[i]) - ord('0') + c) div 10;
  end;
  for i := 1 to n do begin
    write(a[i]);
  end;
  writeln();
end;
```

Задача F. Карта

Рассмотрим произвольную точку местности. Пусть она имеет координаты (x_0, y_0). Тогда ее изображение на карте имеет следующие координаты:

$$(x + \frac{x_0 \cdot a}{W \cdot 100}, y + \frac{y_0 \cdot b}{H \cdot 100})$$

.

Таким образом, для нахождения искомой точки требуется решить два линейных уравнения:

$$x_0 = x + \frac{x_0 \cdot a}{W \cdot 100}$$

$$y_0 = y + \frac{y_0 \cdot b}{H \cdot 100}$$

Решая их, получаем следующие формулы:

$$x_0 = \frac{x}{1 - \frac{a}{100 \cdot W}}$$

$$y_0 = \frac{y}{1 - \frac{b}{100 \cdot H}}$$

Задача G. Шаблоны

Обозначим как S_z множество символов, в которые может быть преобразован символ z $(z = 0, \ldots, 9, a, \ldots, g, ?)$. Обозначим как $|A|$ количество элементов в множестве A.

Пусть $p = p[1], p[2] \ldots p[n]$ и $t = t[1], t[2], \ldots, t[n]$ — заданные шаблоны. Ясно, что количество символов, которые могут стоять на i-ой позиции в строке, которая может быть получена из обоих шаблонов, равно $|S_{p[i]} \cap S_{t[i]}|$.

Ответ на задачу тогда равен $|S_{p[1]} \cap S_{t[1]}| \cdot |S_{p[2]} \cap S_{t[2]}| \cdot \ldots \cdot |S_{p[n]} \cap S_{t[n]}|$.

Задача H. Гонки

Сформулируем поставленную задачу в терминах теории графов: «Задан неориентированный граф. Необходимо найти количество вершинно-простых циклов длины 4».

Пусть нам известна такая матрица a_2, что $a_2[i][j]$ — количество путей длины 2 из вершины i в вершину j. Искомое число циклов равно

$$\frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{(a_2[i][j] \cdot (a_2[i][j]-1))}{2})}{2}$$

.

Осталось вычислить матрицу a_2. Пусть a_1 — матрица смежности заданного графа, то есть $a_1[i][j] = 1$, если между вершинами i и j есть ребро, $a_1[i][j] = 0$, иначе. Иными словами, $a_1[i][j]$ равно количеству путей длины 1 из вершины i в вершину j.

Элементы матрицы a_2 могут быть вычислены по следующей формуле:

$$a_2[i][j] = \sum_{k=1}^{n} a_1[i][k] \cdot a_1[k][j]$$

Это объясняется тем, что в пути длины 2 из вершины i в вершину j есть ровно одна промежуточная вершина. Обозначим ее k. Количество путей длины 2 из i в j, проходящих через k, равно $a_1[i][k] \cdot a_1[k][j]$. Суммируя эту величину по $k = 1 \ldots n$, получаем указанную формулу.

Задача I. Строки

В силу небольших ограничений на длины заданных строк задачу можно решать методом перебора.

Переберём все циклические сдвиги строки b (их не больше ста). После этого попытаемся найти в строке a все подстроки, равные соответствующему циклическому сдвигу. Это можно сделать простым перебором — тогда суммарное время работы составит $O(n^2m)$, или с помощью алгоритма Кнута-Морриса-Пратта — тогда суммарное время работы составит $O(n\,m)$ [2, 3].

Оба этих варианта для заданных максимальных длин строк укладывались в ограничение по времени.

При реализации необходимо было учитывать, что некоторые циклические сдвиги строки b могут быть равны друг другу. Иначе, подстроки строки a, равные таким сдвигам, будут учтены несколько раз.

Для того, чтобы этого избежать, можно, например, воспользоваться массивом типа `boolean`, в котором помечать начальные позиции подстрок строки a, равных некоторому циклическому сдвигу строки b.

Список литературы

[1] Романовский И.В. Дискретный анализ: Учебное пособие для студентов, специализирующихся по прикладной математике и информатике. - 3-е изд., перераб. и доп. - СПб: Невский Диалект; БХВ Петербург, 2003. - 320 с.: ил

[2] Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. - М.: МЦНМО, 1999. - 960 с., 263 ил.

[3] Шень А. Программирование: теоремы и задачи. - М.: МЦНМО, 1995. - 264 с.