Санкт-Петербургский Государственный Политехнический Университет

Задача о пропавших цифрах

Зенцев Ф.
гр. 3057/2

Санкт-Петербург
2010

Содержание

1. Введение

Зачастую при передаче данных случаются потери информации. В данной работе рассмотрим некоторый частный случай потери данных, а именно ***задачу о пропавших цифрах*** (lost digits problem) в контексте выполнения сложения в столбик. Как видно из рисунка 1 речь идет о следующем: было выполнено сложение в столбик, в результате каких-то событий (часто говорят «море смыло цифры, написанные на песке») значения некоторых цифр слагаемых или суммы пропали.

Рисунок 1

2. Постановка задачи

Неформальная постановка задачи ясна: заданы два числа и результат их сложения. Значения некоторых разрядов чисел неизвестны – их необходимо восстановить. Теперь формализуем постановку:

Дано:

$\{a_i\}_{i=1}^n, \{b_i\}_{i=1}^n, \{c_i\}_{i=1}^n$ - значения разрядов чисел *a*, *b*, *c*, записанных в системе счисления с основанием *p*

$N = \{1,2\ldots,n\}$

$$\sum_{i=1}^{n} a_i p^{i-1} + \sum_{i=1}^{n} b_i p^{i-1} = \sum_{i=1}^{n} c_i p^{i-1}$$

$$N_a = \{i \in N \mid a_i = ?\},\ N_b = \{i \in N \mid b_i = ?\},\ N_c = \{i \in N \mid c_i = ?\}$$

Найти:

$$\{a_i\}_{i \in N_a}, \{b_i\}_{i \in N_b}, \{c_i\}_{i \in N_c}$$

3. Описание алгоритма

Как часто случается, задача допускает решение с помощью полного перебора значений для неизвестных разрядов. К сожалению, всего существует $p^{|N_a|+|N_b|+|N_c|}$ комбинаций. Следовательно, в худшем случае, когда решение будет найдено при рассмотрении последней комбинации, временная сложность такого алгоритма будет $\theta(p^{|N_a|+|N_b|+|N_c|})$. Дополнительно, если неизвестно целиком какое-то из слагаемых или сумма, то временная сложность алгоритма, основанном на переборе всех возможных вариантов, примет вид $\theta(p^n)$. Ясно, что при больших n этот способ неприемлем.

Далее будет обсуждаться альтернативный подход, который позволит достичь линейной временной сложности. Идея этого подхода будет основана на простом наблюдении, алгоритм - на хорошо известном алгоритме, поиске в ширину на графе.

3.1. Идея алгоритма

Для того, чтобы понять подход к решению задачи надо, осознать, какие ограничения встречаются при попытке восстановления неизвестных цифр, что вообще мешает случайным образом заполнить неизвестные разряды. Единственное ограничение – это перенос единицы в следующий разряд при превышении суммой цифр текущего разряда основания системы счисления.

Сформулируем идею: для нахождения решения задачи необходимо знать лишь возможен ли перенос единицы в *i+1-ый* разряд при выполнении сложения в *i-ом* разряде.

$$\begin{array}{l} \ldots \overset{+1?}{a_{i+1}} a_i \ldots \\ \underline{\ldots b_{i+1} b_i \ldots} \\ \ldots c_{i+1} c_i \ldots \end{array}$$

Соответственно возникает необходимость в построении такого алгоритма, который позволил бы учитывать данное ограничение для каждого разряда.

3.2. Вспомогательная структура данных

Пусть каждому разряду, то есть каждой тройке (a_i, b_i, c_i) соответствует пара вершин $(s_1^{(i)}, s_2^{(i)}), i \in N$. Пусть тогда G_0 - вполне несвязный граф и $V_{G_0} = \{s_1^{(i)}\}_{i=1}^n \bigcup \{s_2^{(i)}\}_{i=1}^n$. Таким образом мы сопоставили исходным числам *a*, *b* и *c* некоторый набор вершин, рисунок 2 отображает построенное соответствие.

$$\begin{array}{l} a_n a_{n-1} \dots a_2 a_1 \\ \underline{b_n b_{n-1} \dots b_2 b_1} \\ c_n c_{n-1} \dots c_2 c_1 \end{array} \quad \sim \quad G_0:$$

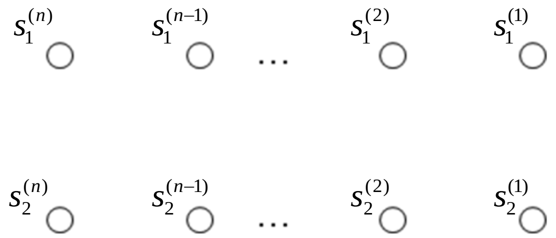


Рисунок 2

Введем теперь конструктивное определение основной структуры данных.

Def. *Орграфом переносов* (directed carry graph) будем называть граф G^* такой, что:

1) $s_1^{(1)} \in V_{G^*}$
2) $(s_j^{(i)}, s_1^{(i+1)}) \in E_{G^*}$, если сложение в i-*ом* разряде возможно выполнить без переноса единицы
3) $(s_j^{(i)}, s_2^{(i+1)}) \in E_{G^*}$, если сложение в i-*ом* разряде возможно выполнить с переносом единицы

Будем считать, что в первый разряд единица не переносится, что и отражает первый пункт определения. Рисунок 3 являет собой пример построенного графа G^*.

$$\begin{array}{l} 23?1? \\ \underline{?59?1} \\ 78?9? \end{array}$$

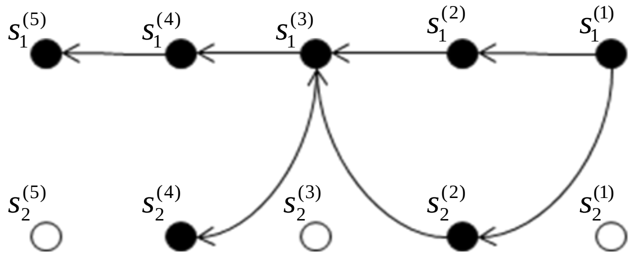
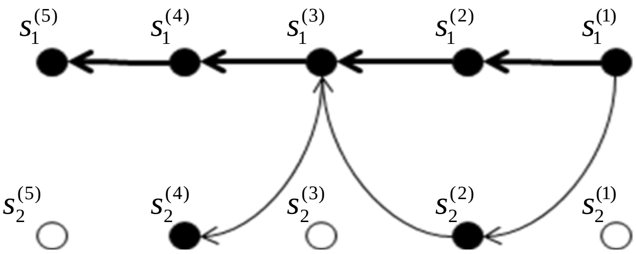


Рисунок 3

3.3. Путь в орграфе переносов

Ключевым шагом алгоритма будет нахождение пути в орграфе «сквозь разряды», а именно пути $F: <\overrightarrow{s_1^{(1)}, s_j^{(n)}}>$, здесь и будет применяться алгоритм поиска в ширину на графе.

$$\begin{array}{l} 23?1? \\ \underline{?59?1} \\ 78?9? \end{array}$$



$$\begin{array}{l} 23010 \\ \underline{55981} \\ 78991 \end{array}$$

Рисунок 4

Теорема 1 Путь F позволит восстановить решение задачи

Если $F = \varnothing$, то есть искомый путь не найден, то и решения задачи не существует. Действительно, это будет означать, что с некоторого разряда переход на следующий (из любого состояния с переносом или же без него) невозможен, что в сущности и означает - решения не найти.

Если же $F \neq \varnothing$, то решение может быть восстановлено проходом против стрелок пути и последовательным восстановлением неизвестных цифр сообразно с вершинами пути.

3.4. Алгоритм

Конструктивное определение орграфа переносов и теорема 1 позволяют сформулировать алгоритм нахождения задачи.

1 $V_{G^*} \leftarrow \varnothing$

2 $\triangleright$ К первому разряду перейдем без переноса

3 $V_{G^*} \leftarrow V_{G^*} \bigcup \{s_1^{(1)}\}$

4 Построение графа G^*

5 $F \leftarrow <\overrightarrow{s_1^{(1)}, s_j^{(n)}}>$ $\triangleright$ Нахождение пути в G^*

6 **if** $F = \varnothing$ $\triangleright$ Путь не найден

7   print "Решения не существует"

8 **else**

9   Восстановление $\{a_i\}_{i \in N_a}, \{b_i\}_{i \in N_b}, \{c_i\}_{i \in N_c}$

Строка 1 инициализирует множество вершин графа переносов пустым множеством, затем в строке 3-4 строится граф. Строки 5-9 отражают теорему 1, откуда и следует корректность работы алгоритма.

Теорема 2 Время работы алгоритма решения задачи о пропавших цифрах составляет $T = \theta(n)$

Строки 1-4 строят граф переносов, ясно что вершин может быть не более, чем содержится в вполне несвязном графе G_0. Нахождение пути с помощью поиска в ширину на графе в строке 5 составляет, как известно из [1], время $O(V_{G^*} + E_{G^*})$. Рассмотрим случай, когда неизвестны целиком слагаемые и сумма, то есть $N_a = N_b = N_c = N$

$$\begin{cases} V_{G^*} \le 2n \\ E_{G^*} \le 2 + (n-2)*4 = 4n - 6 \end{cases}$$

$$V_{G^*} + E_{G^*} = 2n + 4n - 6 = 6n - 6 \in \theta(n)$$

Очевидно, что восстановление неизвестных цифр в строке 9 (проход против стрелок пути) также составляет время $\theta(n)$

Замечания:

В работе детально не рассмотрены два вопроса: как строить орграф переносов и как восстанавливать неизвестные цифры по найденному пути. Ясно, во-первых, что восстановление может быть неоднозначным. Во-вторых, обе эти задачи решаются тривиальным перебором случаев, в зависимости от количества неизвестных в уравнении текущего разряда $a_i +_p b_i = c_i$. Так, если неизвестна только одна цифрах из трех, то она восстанавливается однозначно, с учетом стрелки пути в орграфе – с переносом необходимо перейти или без.

4. Выводы

1. Более тщательный анализ задачи, выявление простых закономерностей позволяет порой существенно снизить временную сложность.

2. Следует помнить и знать хорошо изученные элементарные алгоритмы на графах.

5. Литература

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. «Алгоритмы: построение и анализ», М: «Вильямс», 2005
2. Новиков Ф.А. «Дискретная математика для программистов: Учебник для вузов. 3-е изд.», СПб: «Питер», 2008