**Дисциплина «Алгоритмы решения прикладных задач»**

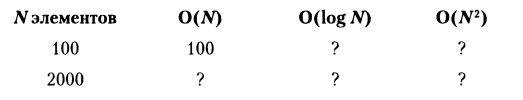
**Рабочая тетрадь 3.**

**Оптимизация на основе О-нотации. Алгоритмы длинной арифметики: сложение, вычитание, умножение, деление**

|  |  |
| --- | --- |
| **Теоретический материал** | |
| Рассматриваемые алгоритмы применимы, прежде всего, к длинным и очень длинным целым числам (состоящими, например из 100 или 1000 цифр). Такие числа не могут быть записаны в переменную стандартных типов (int, long и т.п.). Поэтому программная реализация арифметических алгоритмов будет предполагать работу с массивом, каждый элемент которого будет представлять собой цифру в соответствующем разряде длинного числа. Массив можно создавать статически или динамически. Предпочтительный вариант – динамическое создание массивов (на языке C++ с помощью указателей). При этом в отдельной переменной необходимо хранить размер массива (количество цифр в длинном числе).  *Пример 1 демонстрирует создание динамического массива для числа, содержащего задаваемое с клавиатуры количество цифр. Массив заполняется* ***случайными*** *цифрами от 0 до 9 (старший разряд не может быть нулевым в этом случае). При этом в нулевом элементе массива содержится младший разряд, а в последнем – старший. То есть на печать массив нужно выводить наоборот, чтобы число отображалось правильно.*  *Пример 2 демонстрирует сравнение двух больших чисел. Это необходимо, например, при реализации операции вычитания для двух больших чисел. Генерация большого числа реализована через функцию*  **Сложение и вычитание**  Базовые алгоритмы сложения, вычитания, умножения и деления чисел – хорошо известные алгоритмы выполнения действия «в столбик». Вместе с тем вся машинная арифметика сводится к одной операции сложения двух двоичных чисел. В этой связи, отдельного рассмотрения заслуживают задачи ускорения операции сложения. Основная идея ускорения состоит в раздельном рассмотрении поразрядного сложения и переносов. В двоичной системе это реализуется следующим образом, см. рисунок ниже, пример    ***Рисунок 1 – Способ ускорения операции сложения***  Здесь в качестве промежуточных значений используются два числа, промежуточная сумма и число переносов. Процесс завершается тогда, когда число переносов полностью обнуляется. На первом шаге суммируются исходные числа, на каждом последующем – промежуточная сумма и число переносов предыдущего шага. Промежуточная сумма формируется согласно следующему правилу: *k-ый разряд суммы равен 0, если равна 0 или 2 сумма k-ых разрядов слагаемых, k-ый разряд суммы равен 1, если равна 1 сумма k-ых разрядов слагаемых.*  Число переносов формируется согласно следующему правилу: *самый младший разряд числа всегда равен 0. k-ый разряд равен 1, если сумма (k–1)-ых разрядов равна 2. В иных случаях разряд числа переносов равен 0.*  **Умножение**  Умножение столбиком можно назвать естественным алгоритмом, в том смысле, что столбиковое умножение непосредственно опирается на таблицу умножения цифр и определение операции умножения через сложение. Естественные идеи такого рода хороши для реализации, так как предполагают только некоторые технические сложности и не более того, но вряд ли так можно получить максимум эффективности.  **Метод Карацубы**    При этом если числа являются разной длины или имеют нечетную длину, то целесообразно добавить старший «нулевой разряд». Например, если даны числа 37656 и 6567863, то их следует рассматривать в таком виде:  00037656 = 0003\*104 + 7656;  06567863 = 0656\*104 + 7863;  Будем рассматривать произведение двух чисел в виде (ax+b)(cx+d), где x=10k (в примере выше k=4). Имеет место следующая цепочка равенств    Для наших двух чисел имеем:  (0003\*104 + 7656)\* (0656\*104 + 7863)=  0003\*0656\*108+((0003+7656)\*(0656+7863)-0003\*0656-7656\*7863)\*104+7656\*7863  В этом выражении произведения 0003\*0656 и 7656\*7863 повторяются дважды, а всего нужно посчитать 3 произведения четырехразрядных чисел:   1. 0003\*0656 2. 7656\*7863 3. (0003+7656)\*(0656+7863) = 7659\*8519   Каждое произведение четырехразрядных чисел можно разбить на произведения двухразрядных, а произведения двухразрядных – на произведения одноразрядных чисел.  Пример для умножения четырехразрядных чисел методом Карацубы:    **Деление. Алгоритм с использованием метода половинного деления**  Пусть некоторое время нас интересует только частное. Частное можно представить как неизвестную величину в уравнении вида:  Как известно, такие уравнения можно решать методом половинного деления. Здесь *f = A–Bp*, левая граница отрезка, содержащего корень, есть 0 и правая граница равна делимому A. Решением уравнения будет частное *p.* Далее остаток находится просто. Если частное получено с точностью до целого, то остаток легко определяется из формулы *A = Bp + q ⇔ q = A − Bp*  Вместо метода половинного деления можно использовать и другие методы: метод хорд или касательных | |
| **Пример 1** | |
| ***Задача:*** | |
|  | Написать на языке C++ программу, в которой создается динамический массив для хранения длинного целого числа (более 10 цифр). Размер массива задается с клавиатуры. Цифры генерируются случайным образом. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Пример 2** | |
| ***Задача:*** | |
|  | *Написать на языке C++ программу для* *сравнения двух больших чисел (более 10 цифр)* |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
|  |  |
| **Задание 1** | |
| ***Задача:*** | |
|  | Следующая функция находит наибольшее число в массиве, но имеет сложность О(N2). Перепишите эту функцию так, чтобы ее сложность была O(N): |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 2** | |
| ***Задача:*** | |
|  | Написать программу для сложения и вычитания двух больших чисел в столбик. Для представления чисел использовать массив, в котором будут храниться цифры числа. При реализации процедуры вычитания предпочтительно из большего вычитать меньшее. Для этого первоначально определять, какое из двух чисел больше |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 3\*** | |
| ***Задача:*** | |
|  | Написать программу для реализации ускоренного сложения чисел в двоичной системе, аналогично примеру: |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 4** | |
| ***Задача:*** | |
|  | Написать программу для умножения двух больших чисел столбиком. Обратите внимание на количество цифр в произведении: их скорее всего будет больше, чем в исходных числах |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 5\*** | |
| ***Задача:*** | |
|  | Написать программу для умножения двух больших чисел методом Карацубы. Код программы объединить с кодом для задания 4, чтобы иметь возможность сравнить результат умножения одних и тех же чисел столбиком и методом Карацубы |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Задание 6** | |
| ***Задача:*** | |
|  | Написать программу для нахождения частного и остатка от деления двух чисел методом половинного деления |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |

**Дополнительные вопросы для текущего контроля**

1. Замените вопросительные знаки в таблице ниже количеством шагов алгоритма при заданном количестве элементов данных

****

1. Если во время обработки массива с помощью алгоритма О(N2) мы выясняем, что при этом он выполняет 256 шагов, то каков размер массива?
2. С помощью О-нотации определите временную сложность следующей функции, которая находит наибольшее произведение любых пар чисел в заданном массиве:

