
Задачи на работу со списками

А. Г. Фенстер, `fenster@fenster.name`

16 апреля 2010 г.

Конспект семинара №10 по программированию для студентов 1 курса ММФ НГУ. Помните, что чтение конспекта не делает посещение соответствующего семинара необязательным! О найденных ошибках и неточностях, пожалуйста, пишите мне по адресу `fenster@fenster.name`. Также буду рад получить любые комментарии по поводу этого текста.

1 Решение задач на работу со списками

Приведённые задачи вряд ли имеют какую-то практическую ценность, однако позволяют «прочувствовать» принципы работы с односвязными списками.

Задача. Выполнить разворот списка за один проход по нему.

Для решения задачи предлагается сначала понять, как будет выглядеть тело цикла. Установив указатели `p`, `q` и `r` на три подряд идущих элемента списка, мы «разворачиваем стрелку» между `p` и `q` (`q->next = p;`), не теряя при этом «хвост» списка `r`.

В результате получим следующий фрагмент кода (чтобы разобраться с ним, нарисуйте на бумаге небольшой список из трёх-четырёх элементов и выполните его вручную).

```
struct item *p, *q, *r;

if (!head)
    return;
if (!head->next)
    return;

p = head;
```

Задачи на работу со списками

```
q = head->next;

while (q)
{
    r = q->next;
    q->next = p;
    p = q;
    q = r;
}

head->next = NULL;
head = p;
```

Задача. Вставьте элемент перед каждым элементом списка, удовлетворяющим некоторому условию.

Мы знаем, как вставить элемент *после* данного, но не можем вставить элемент *перед* данным: для этого необходимо исправлять указывающую на него «стрелку». Для решения такой задачи можно предложить проход по списку двумя указателями (с запоминанием предыдущего):

```
struct item *p = head, *q = NULL;

while (p)
{
    if (элемент p искомый, то есть необходимо вставить новый элемент перед ним)
        break;
    q = p;
    p = p->next;
}
```

После цикла достаточно разобрать два варианта: если q равно $NULL$, необходимо выполнить вставку элемента в начало списка, в противном случае необходимо вставить элемент после q .

Аналогичным образом решается практическое задание о сортировке последовательности чисел при помощи списка: при вставке нового элемента в список движемся по нему от начала, пока значения просматриваемых элементов меньше нового, и вставляем элемент перед текущим, как только значение текущего элемента окажется больше.

Задача. Слейте два упорядоченных списка в один, используя те же элементы (меняя значения полей `next`).

Задача. «Задача Иосифа»: смоделируйте на циклическом списке известную задачу. N человек выстроены в круг, на каждом шаге из круга

выходит каждый K -й (отсчёт продолжается далее по кругу). Определить, кто из людей останется последним. При решении на циклическом списке обратите внимание на два момента: во-первых, удобнее отсчитывать $K - 1$ человека, чтобы затем выполнять простую операцию удаления следующего за данным элемента; во-вторых, после удаления каждого элемента необходимо убедиться, что не был удалён элемент списка, на который указывает переменная `head`, в противном случае вычисление условия окончания (остался один элемент, `head->next == head`) вызовет ошибку.

2 Проверочное задание («пятиминутка»)

Тема: работа со списками (см. [задание](#)).