

Задача 8. Список с индексами

Источник:	повышенной сложности*
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

У связанного списка есть серьёзная проблема: в отличие от массива в нём нельзя быстро получить k -ый по порядку элемент. Это можно сделать лишь перебором узлов списка за $O(k) \approx O(N)$ времени. В данной задаче предлагается ускорить поиск узла по индексу.

Рассмотрим односвязный список. Пусть в списке помимо обычных узлов (“маленьких” узлов) есть ещё немного особенных узлов (“больших” узлов). В любом узле списка нужно хранить значение узла и указатель на следующий по порядку элемент. В каждом большом узле списка предлагается дополнительно хранить указатель на следующий **большой** узел списка и расстояние до него. Под расстоянием здесь понимается то, сколько переходов вперёд нужно сделать из одного узла, чтобы попасть по второй узел.

Дополнительные ссылки в больших узлах позволяют быстрее найти k -ый элемент, так как можно проскакивать сразу по много узлов, изначально проходя исключительно по большим узлам. Можно заметить, что если доля больших узлов в списке примерно $\frac{1}{B}$, то найти k -ый элемент можно примерно за $O(\frac{k}{B} + B)$. (**Вопрос:** как лучше выбрать B ?)

При добавлении элементов важно поддерживать указанную выше структуру. Вставку узла будем выполнять по индексу, на котором должен стоять новый элемент. При этом сначала нужно решить, будет ли новый узел большим — это можно делать случайным образом, так чтобы поддерживать желаемую долю больших узлов. Затем нужно найти по индексу последний большой и малый узлы перед тем местом, куда нужно вставить новый узел. Наконец, можно вставить новый узел, корректно обновив все ссылки и расстояния так, чтобы сохранилась структура списка.

Предлагается реализовать эту структуру данных, и обработать с её помощью серию из N запросов двух типов (в описаниях L — текущее количество узлов в списке):

0. Вставить на k -ую позицию новый узел с заданным значением V . Гарантируется, что во всех запросах k лежит в пределах от 0 до L .
1. Вывести значение узла, находящегося на k -ой позиции в списке. Гарантируется, что во всех запросах k лежит в пределах от 0 до $L - 1$.

Решение в целом должно работать за время $O(N\sqrt{N})$.

Формат входных данных

В первой строке задано целое число N — общее количество запросов ($1 \leq N \leq 10^5$). В каждой из следующих N строк дан один запрос. Запрос вставки задаётся тремя целыми числами 0 k V , а запрос на вывод элемента задаётся двумя целыми числами 1 k . Все числа V в списке целые, по модулю не превышают 10^9 .

Формат выходных данных

Для каждого запроса на вывод (типа 1) нужно вывести значение k -ого узла в списке.

Пример

input.txt	output.txt
10	2
0 0 7	5
0 0 5	3
0 1 3	7
0 0 2	5
1 0	
1 1	
1 2	
1 3	
0 1 -5	
1 2	

Комментарий

Несмотря на формально лучшую асимптотику, полученная структура данных будет обрабатывать заданные запросы не быстрее простого массива.