

## Задача 9. Список с индексами++

Источник:	повышенной сложности*
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	3 секунды*
Ограничение по памяти:	разумное

Данная задача посвящена дальнейшему улучшению структуры данных из задачи “Список с индексами”.

Вместо деления узлов лишь на “большие” и “маленькие”, припишем каждому узлу “высоту”. Высота определяет, сколько в узле хранится ссылок на следующие элементы. В предыдущей задаче список имел только два уровня: “маленькие” узлы имели высоту 1, а “большие” — высоту 2. В этой задаче предлагается завести 20 уровней (на самом деле примерно  $\log_2 N$ ), распределяя высоты от 1 до 20.

В узле высоты  $h$  нужно хранить массив из  $h$  указателей вперёд вместе с расстояниями.  $k$ -ый указатель смотрит на следующий в списке элемент, высота которого хотя бы  $k$  (здесь полагаем, что  $k$  считается с единицы). Для идеальных результатов узлов высоты  $k$  должно быть примерно  $2^{-k}$  по доле. Этого можно достичь, если при добавлении каждого нового узла бросать монетку до тех пор, пока не выпадёт орёл, а высоту устанавливать как количество выполненных бросков.

При правильной реализации этой структуры можно выполнять поиск по индексу и добавление узлов примерно за  $O(\log N)$ . Такая структура называется “список с пропусками” / “skip lists”: ссылка.

В данной задаче нужно обработать  $N$  таких же запросов, как в задаче “Список с индексами”, но теперь за время  $O(N \log N)$ .

### Формат входных данных

В первой строке задано целое число  $N$  — общее количество запросов ( $1 \leq N \leq 10^6$ ). 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	