SmoothSort

**Плавная сортировка** — алгоритм сортировки, модификация сортировки кучей. В худшем случае сортировка работает за время **O(NlogN).** Преимущество плавной сортировки в том, что её время работы приближается к **O(N),** если входные данные частично отсортированы, в то время как у сортировки кучей время работы не зависит от состояния входных данных.

1. Рассмотрим случай, когда время работы приближается к **O(N)**.

(Если видео не запустилось автоматически, то вы можете посмотреть видео из папки)

Как мы видим, экономия происходит за счёт просейки. В почти упорядоченных данных просейка опускается неглубоко в дереве. В изначально рандомных данных просейка обходится дороже, так как часто опускается в своей куче до самого последнего уровня

1. Теперь рассмотрим общую временную сложность **O(NlogN)**.

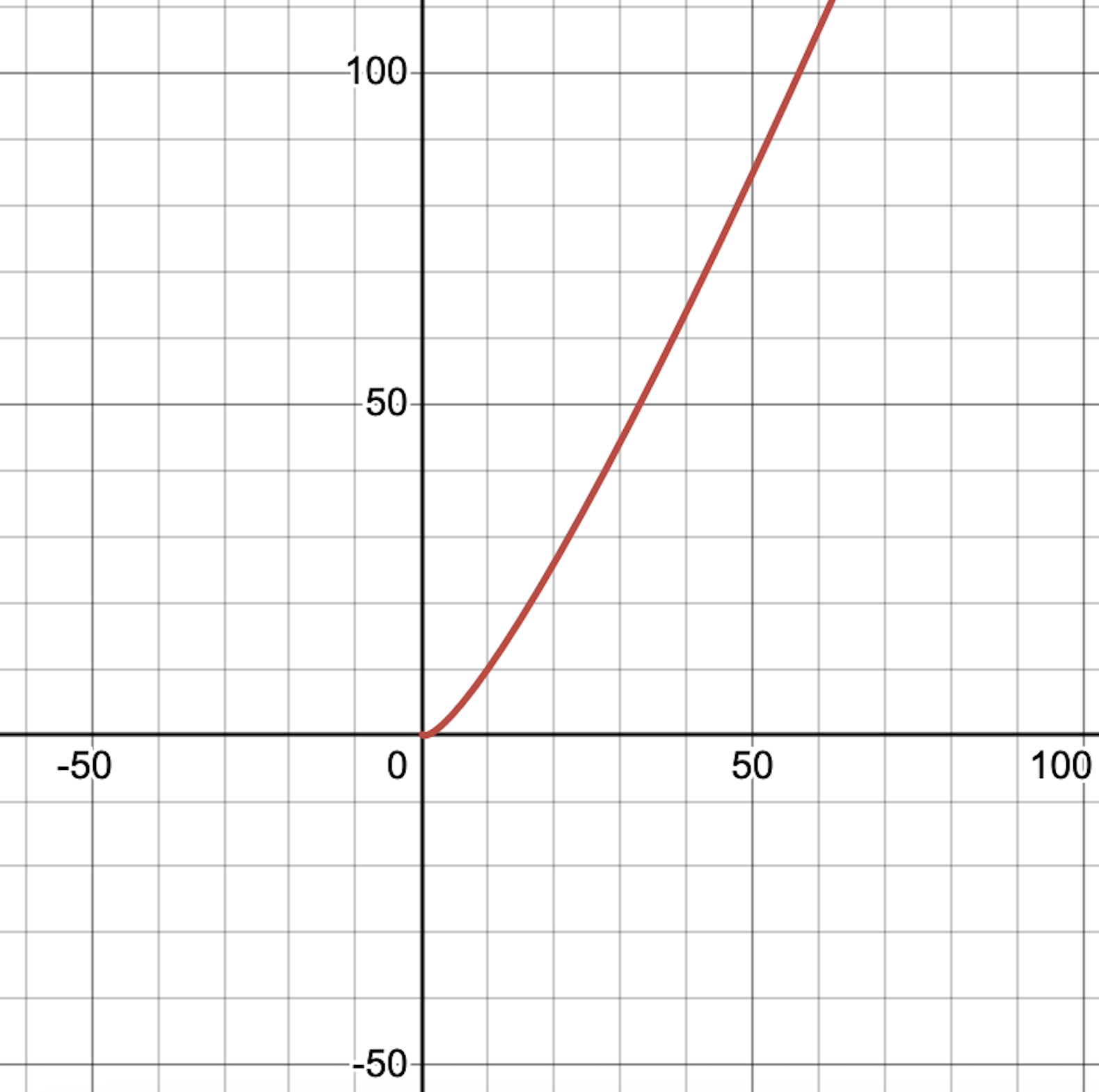
На первом этапе мы перебираем **N** элементов, добавляя его в уже имеющиеся слева кучи. Само добавление в кучу обходится в **O(1)**, но затем для кучи нужно сделать просейку. В упорядоченных данных неглубокая просейка часто обходится **O(1)** для одного добавляемоего в кучу элемента. В неупорядоченных данных просейка для каждого добавления обходится в **O(logN)**, так как как из-за рандома просейке приходится проходить уровни дерева часто до самого низа.

Поэтому, на первом этапе наилучшая сложность по времени:  
для почти упорядоченных данных — **O(N)**,  
для рандомных данных — **O(NlogN).**  
  
Для второго этапа ситуация аналогичная. При обмене очередного максимума опять нужно просеять кучу, в корне которой он находился. И показатели просейки для упорядоченных и неупорядоченных данных будут различные.  
  
На втором этапе наилучшая сложность по времени такая же как и на первом:  
для почти упорядоченных данных — **O(N)**,  
для случайных данных — **O(NlogN)**.  
  
Складывая временны́е сложности для первого и второго этапа:  
для почти упорядоченных данных — **O(2\*N)** = **O(N)**,  
для случайных данных — **O(2\*NlogN)** = **O(NlogN)**.  
  
В общем, худшая и средняя временна́я сложность для плавной сортировки — **O(NlogN)**

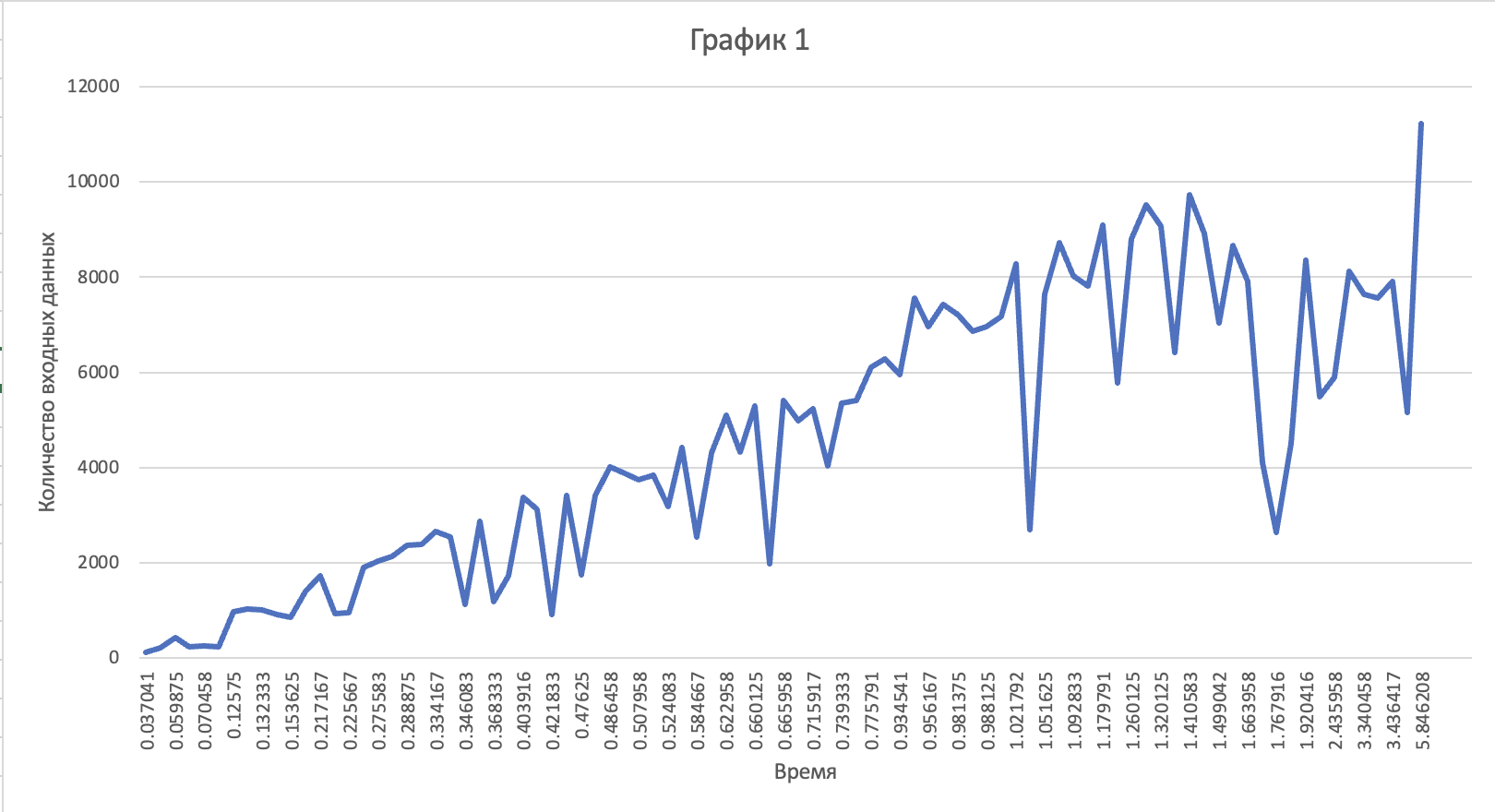
**Примечание по сложности:** требуется **O(NlogN)** дополнительной памяти для хранения длин куч в последовательности. Однако с помощью некоторых модификаций расходы на дополнительную память можно сократить до **O(1).**

**Сравниваем графики**

График **O(NlogN):**



**Наш** график:



Как мы видим на нашем графике есть ямы, которые обусловлены особенностью данной сортировки, а именно частичной готовой сортировкой во входных данных (в данном случае сложность **O(N)**).