HashMap — это класс, который реализует интерфейс Map что подразумевает хранение данных в виде пар ключ/значение. В своей структуре использует массив из Связных списков(Node) где каждый элемент списка хранит ссылку на следующий элемент! но когда список в массиве достигает своего порога, то Java использует бинарное дерево. Это связано со скорость выполнения операций. сбалансированное дерево - реализует алгоритм бинарного поиска O (Log n). **Не гарантирует порядок.**

**Увеличение массива:**

Длина массива(capacity) в map по умолчанию равна 16, но в отличие от ArrayList здесь увеличение массива происходит не тогда, когда он будет заполнен, а когда будет заполнен какой-то процент массива, так называемый load Factor. Если load Factor 0.60 то, когда массив будет заполнен на 60% будет создан новый массив.

HashMap - хранит в себе пару Ключ Значение. И ключ, и значение могу быть записаны как null

Map - не синхронизован

**Как устроен HashMap:**

Когда мы хотим положить элемент в map, то генерируется hashcode, далее идет вычисление индекса путем подсчета остатка от деления или побитовое умножения, после чего, объект помещается в полученный индекс.

1:(& это побитовое умножение, n это длина массива: hashcode & (n-1) ---> после этого:)

2:(hashCode / n-1) % n;

Вычисляем хеш у ключа и далее вычисляем индекс по формулам, что я написал выше.

Скорость выполнения:

1. Поиск элементов в Map в лучшем случае будет за O (1) в худшем за O(n)
2. если в ячейке лежат много элементов, то поиск будет за O(n)
3. Получение или установка значения осуществляется за константное время O (1)

LinkedHashMap - гарантирует порядок, под капотом имеет связной список