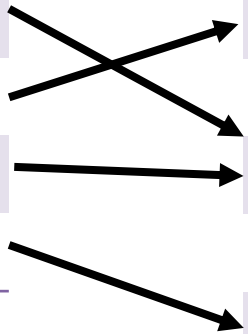
HashMap

Keys	Values
A	11111
DD	2222
C	33333
S	4444



Hash(Keys)	Values
1	2222
2	11111
3	33333
4	
5	4444

Search **O(1)**
Insertion **O(1)**
Deletion **O(1)**

ConcurrentHashMap

- Uses locks inside the usual methods: **get**(), **put**()
- This means that when you use the HashMap in a multi-threaded program you don’t need to remember to lock or what locks you used
- Has special methods such as **putIfAbsent**()

ConcurrentHashMap – under the hood

- If we use one lock for accessing the HashMap: 2 threads can not modify the hash map at the same time.
- If we use one lock for every element in the HashMap: We will need a lot of locks (difficulties with insertion/deletion)
- Store the data in groups and have a limited number of locks (parallelism level). Best results when number of locks is equal to number of threads.

Hash(Keys)	Values	
1	2222	Lock A
2	11111	
3	33333	Lock B
4		
5	4444	Lock C