Coursework

Java_2

Sang Haotian 16722022

Random Name & Age

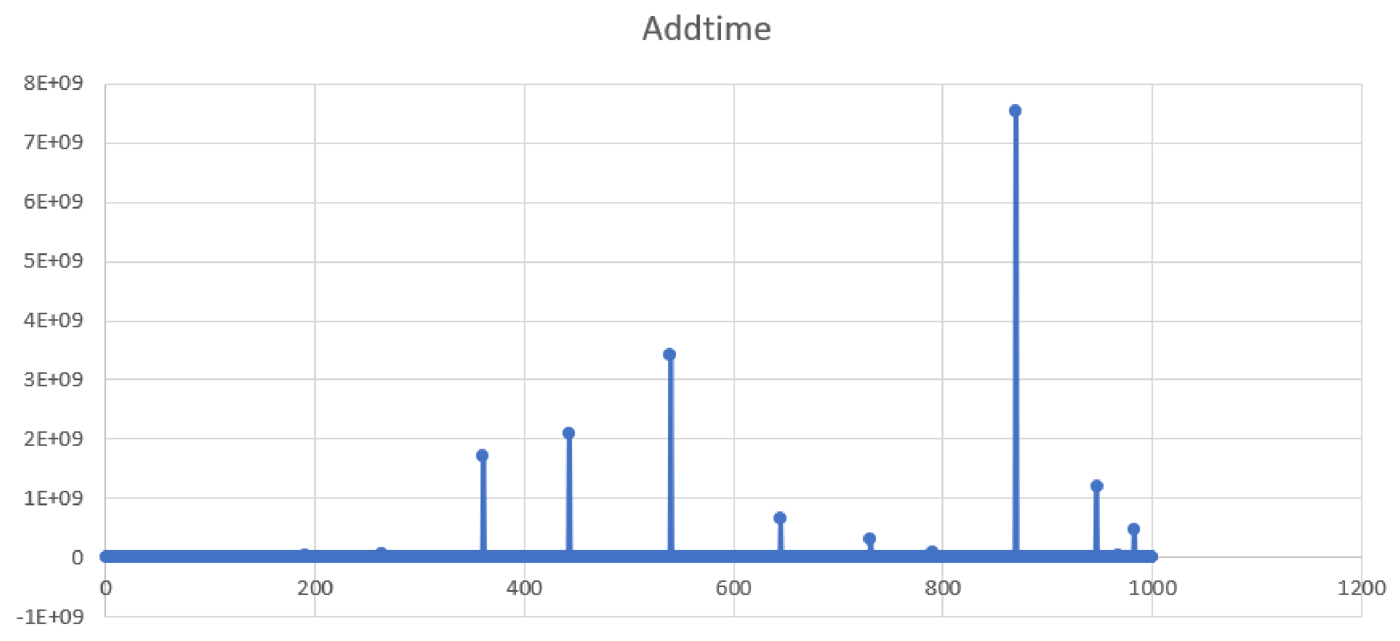
```
String randomName;
String firstName = "",secondName = "";

firstName= firstName+
(char)(int)(Math.random()*26+65);
for(int i=0; i<nameLength; i++) {
secondName = secondName +
(char)(int)(Math.random()*26+97);
}

randomName =  firstName.concat(secondName);
```

```
int age ;
age = (int)(1+Math.random()*(100));
return age;
```

```
String name = "Tom";
name = name + nameNum;
nameNum+=1;
```

ArrayList_Add

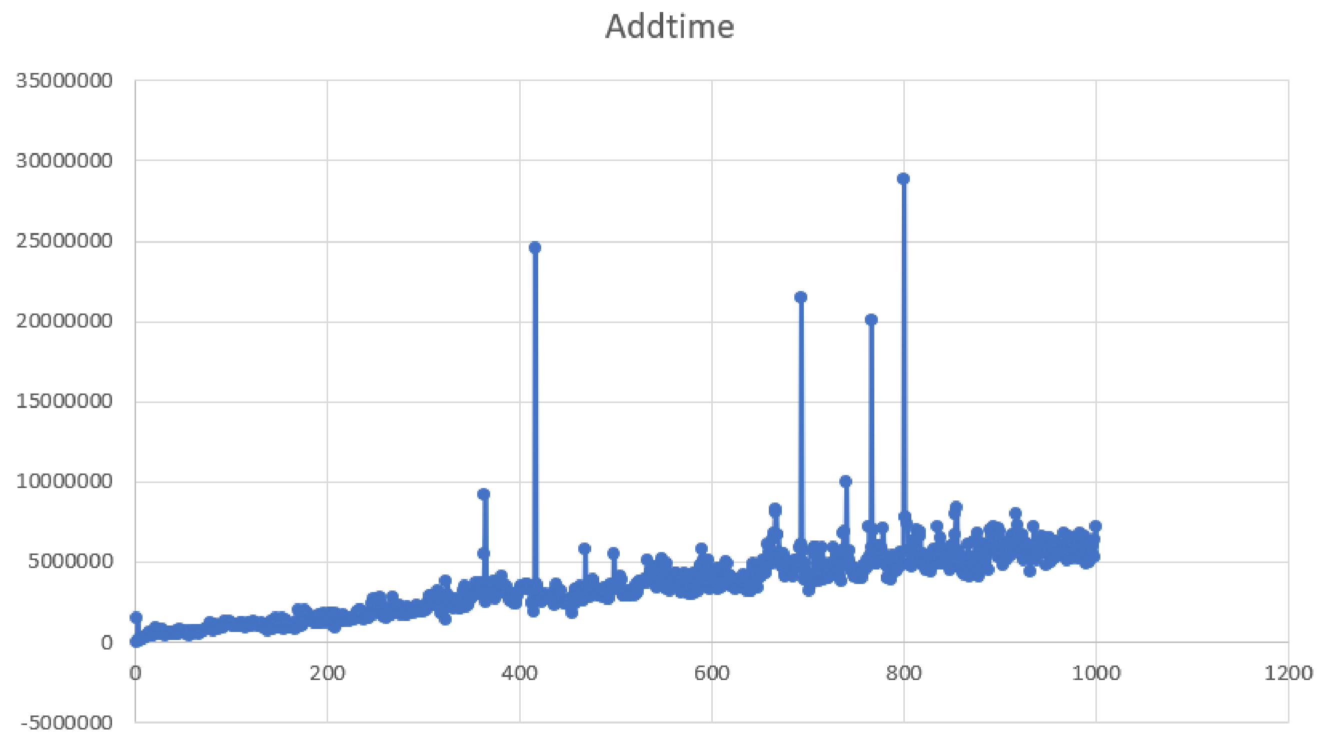
Add to the tail directly, so time complexity O(1)



Temporal complexity: O(1)

- Add 1 million items
- Record 1000 points from whole procedure
- Distance between each point is 1000
- ns

ArrayList_Add

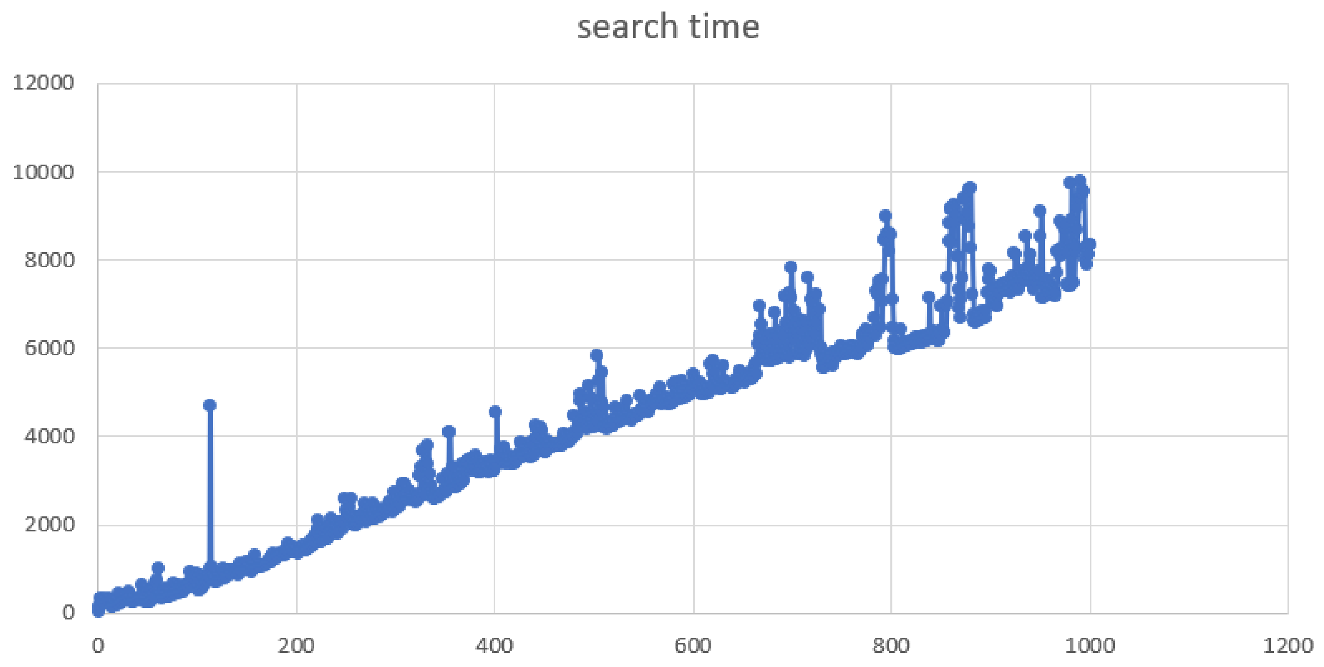


Temporal complexity: O(1)

- Add 100 million items
- Record 1000 points from whole procedure
- Distance between each point is 1000
- ns

ArrayList_Search

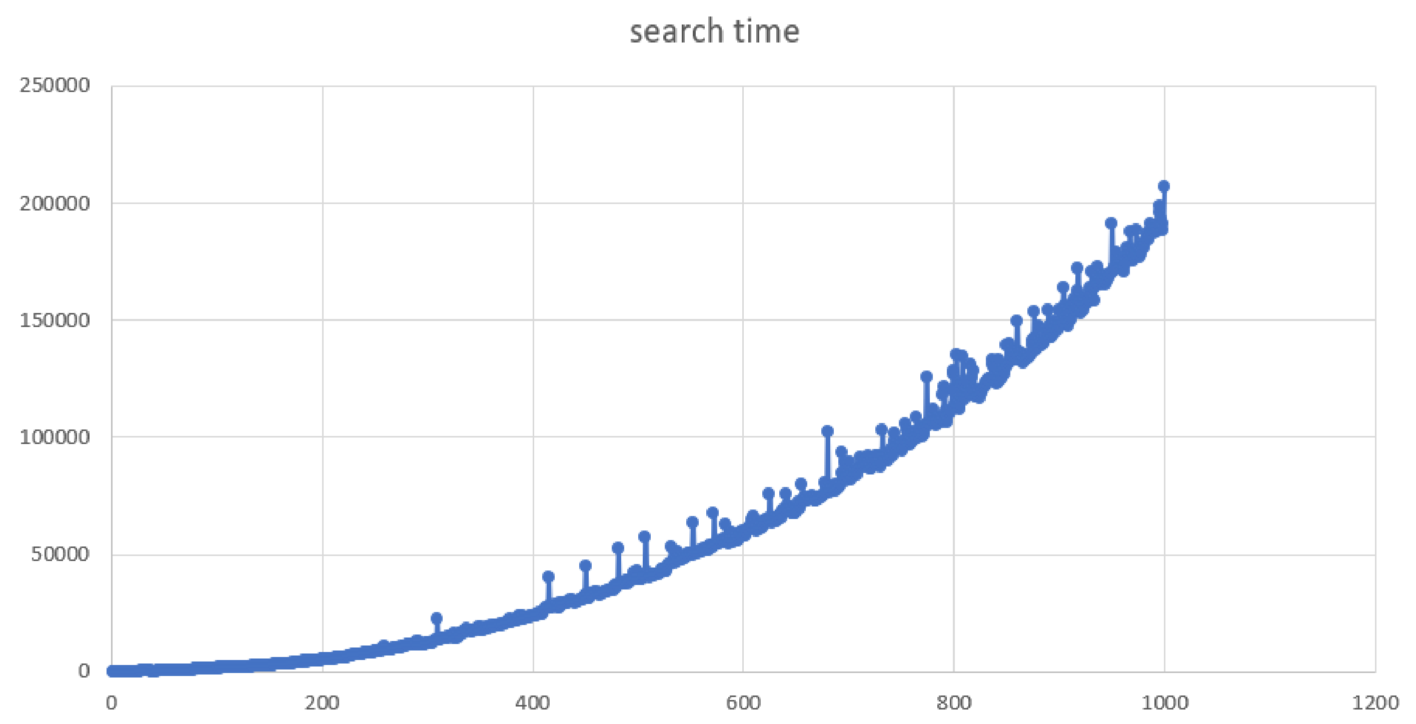
Iterate the whole list to find, so time complexity O(n)



Temporal complexity: O(n)

- Search from1 million items
- Record 1000 points from whole procedure
- Distance between each point is 1000
- ums

ArrayList_Search

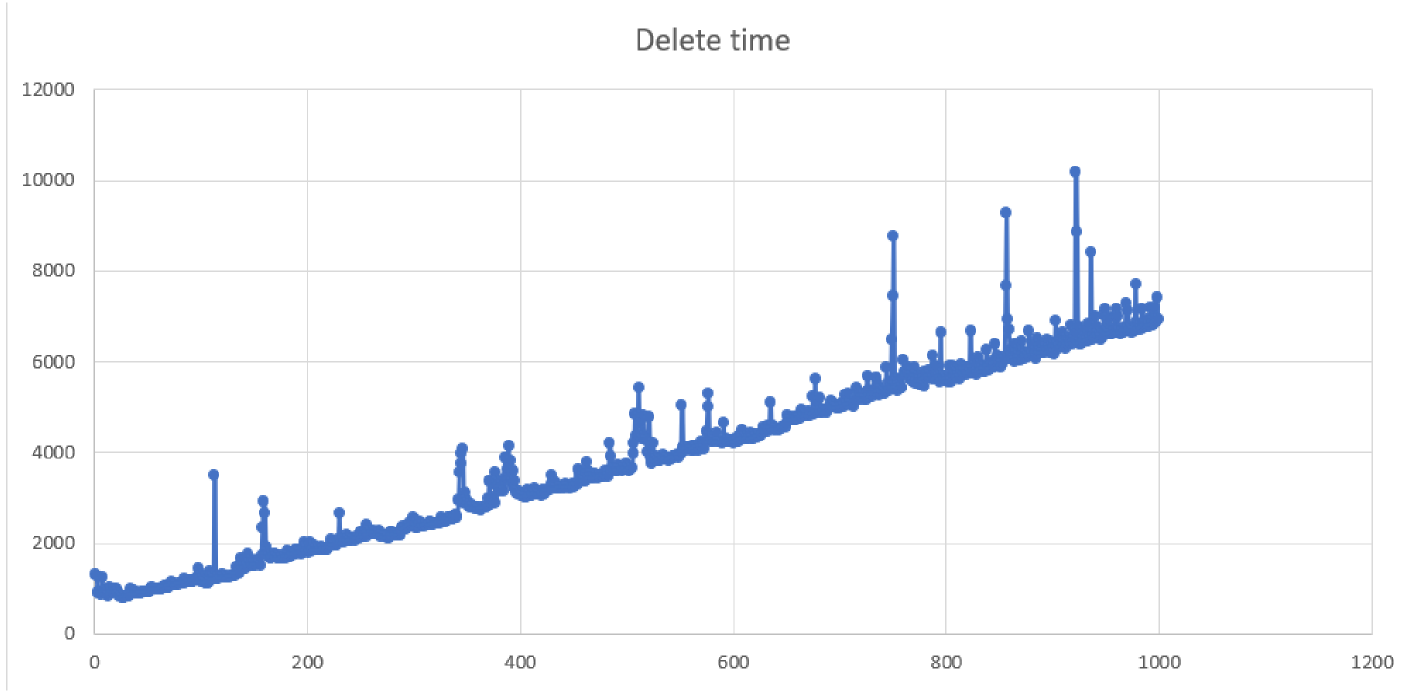


Temporal complexity: O(n)

- Search from1 million items
- Record 100 points from whole procedure
- Distance between each point is 100
- ums

ArrayList_Delete

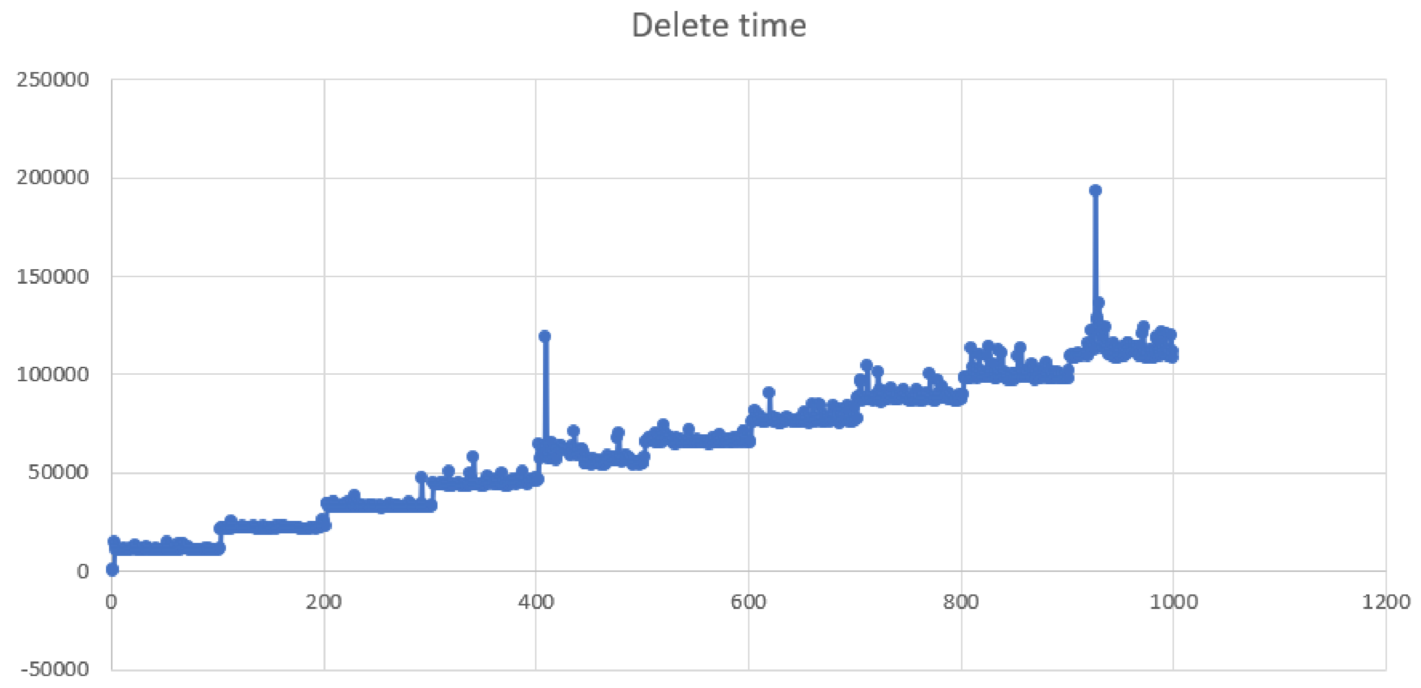
Deletes the specified element, and the subsequent element moves forward, so time complexity O(n)



Temporal complexity: O(n)

- Delete from1 million items
- Record 1000 points from whole procedure
- Distance between each point is 1000
- ums

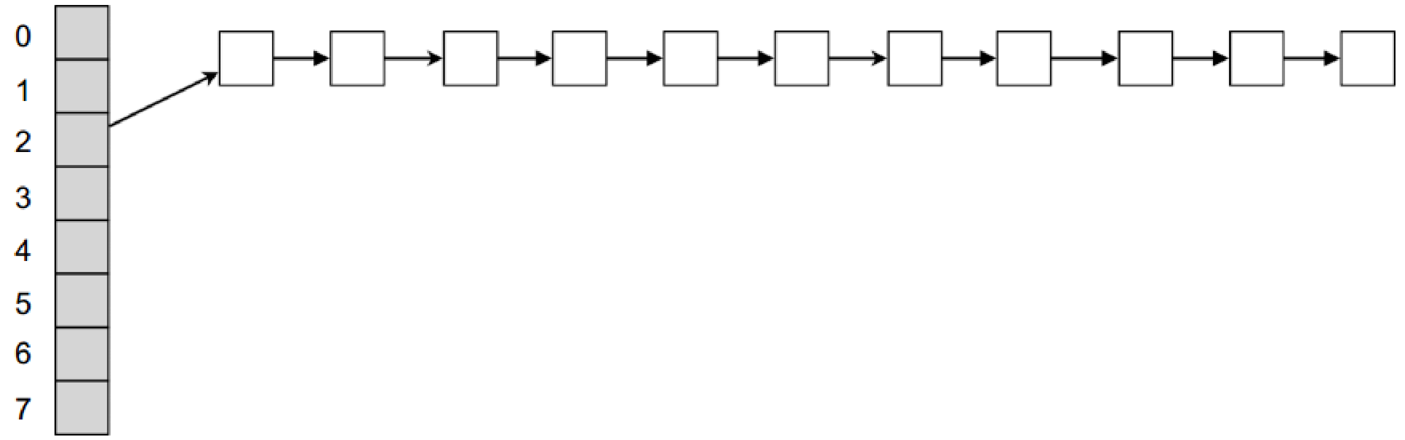
ArrayList_Delete



Temporal complexity: O(n)

- Delete from1 million items
- Record 1000 points from whole procedure
- Distance between each point is 100
- ums

Hashmap

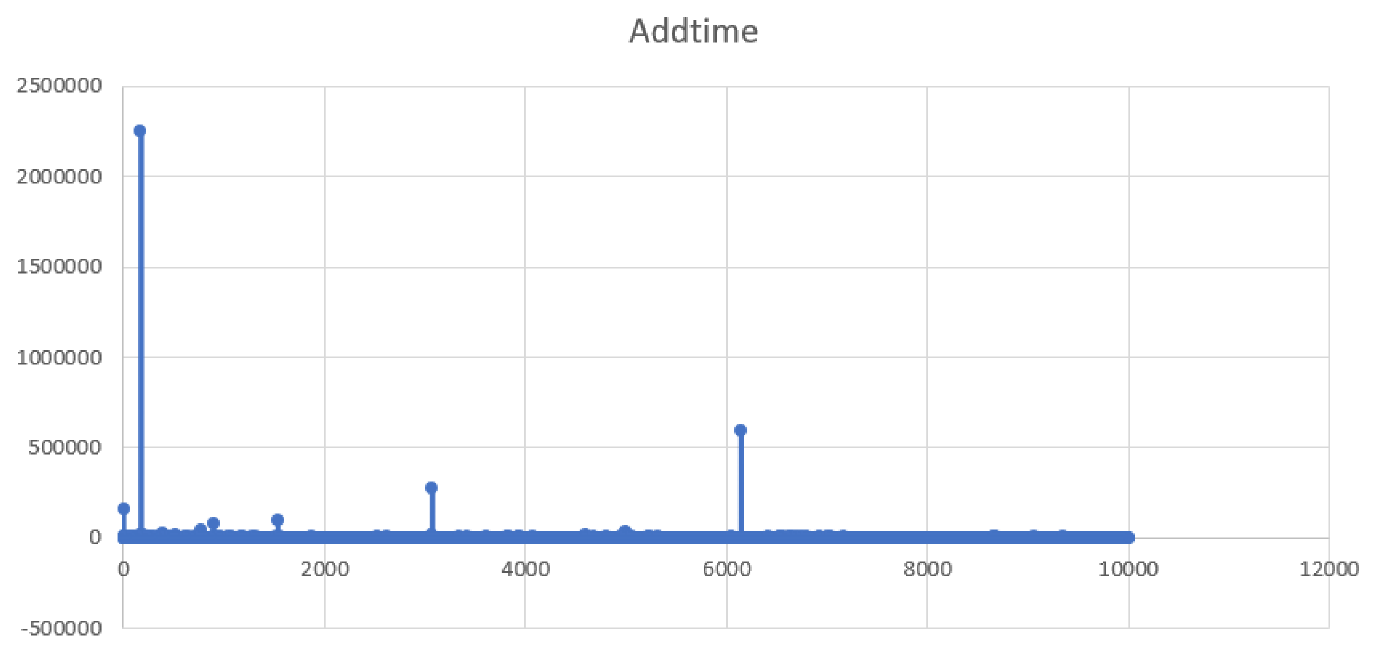


Hash retrieval :

1. Use hashing function to compute index;
2. Search in array element to see if there is a match

Hashmap_Add

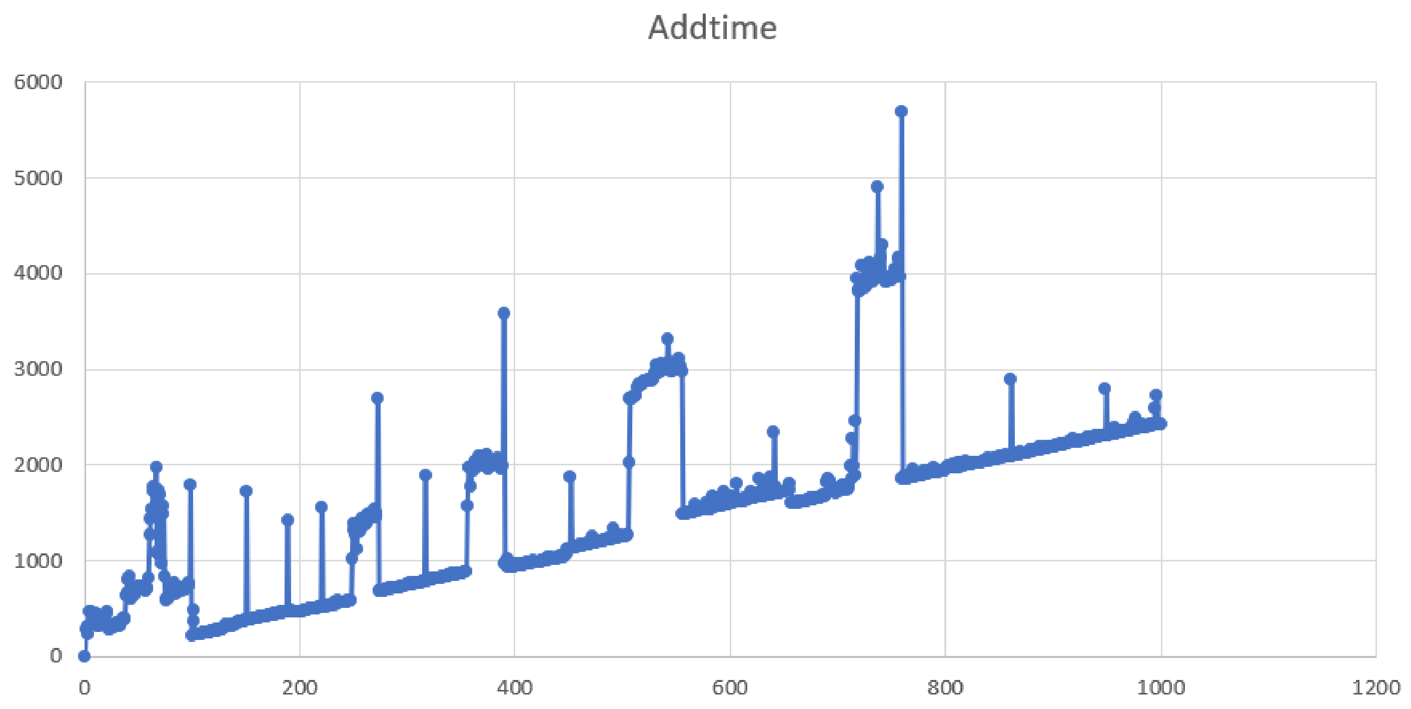
Use the internal key to add, so time complexity O(1)



Temporal complexity: O(1)

- Add 1 million items
- Record 10000 points from whole procedure
- Distance between each point is 1000
- ums

Hashmap_Add

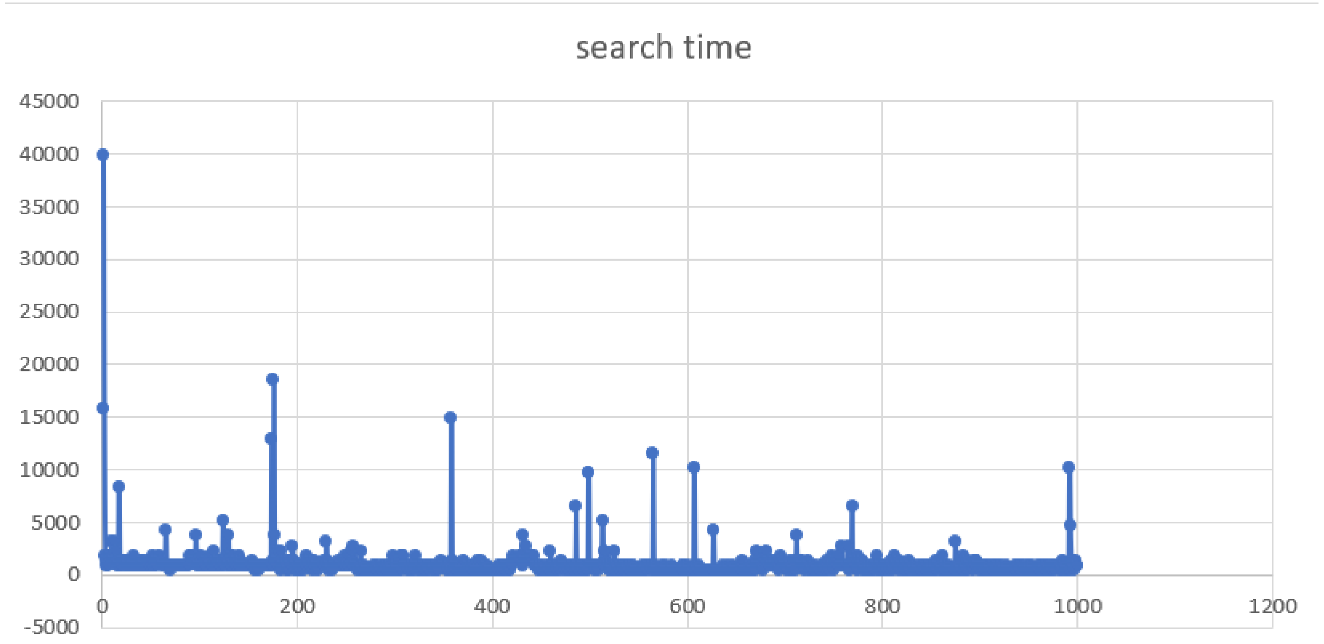


Temporal complexity: O(1)

- Add 1 million items
- Record 1000 points from whole procedure
- Distance between each point is 1000
- ums

Hashmap_Search

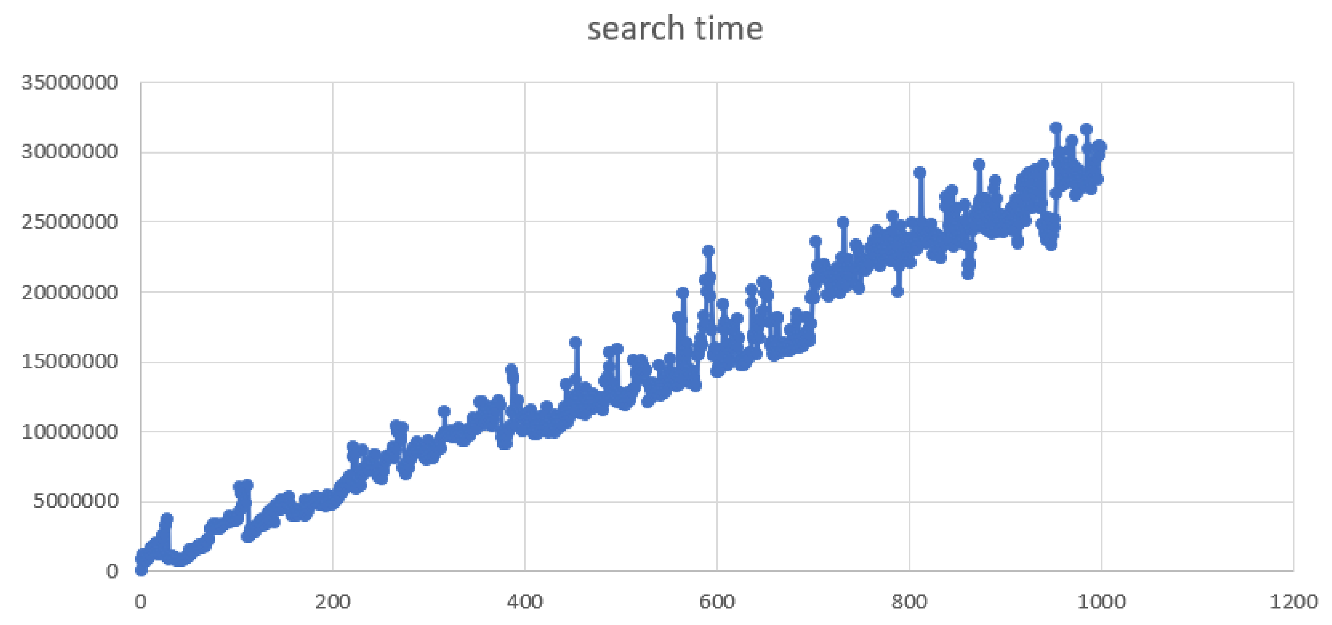
Use the internal key to search, so time complexity O(1)



Temporal complexity: O(1)

- Search from1 million items
- Record 10000 points from whole procedure
- Distance between each point is 1000
- ns

Hashmap_Search

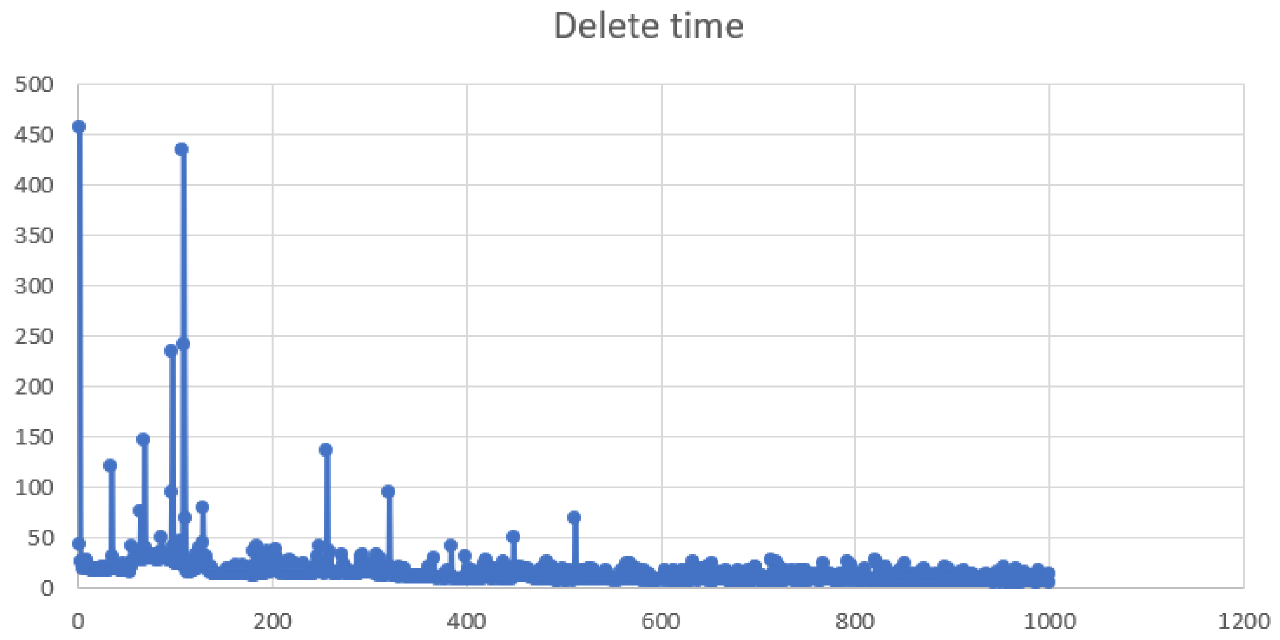


Temporal complexity: O(1)

- Search from1 million items
- Record 1000 points from whole procedure
- Distance between each point is 100
- ns

Hashmap_Delete

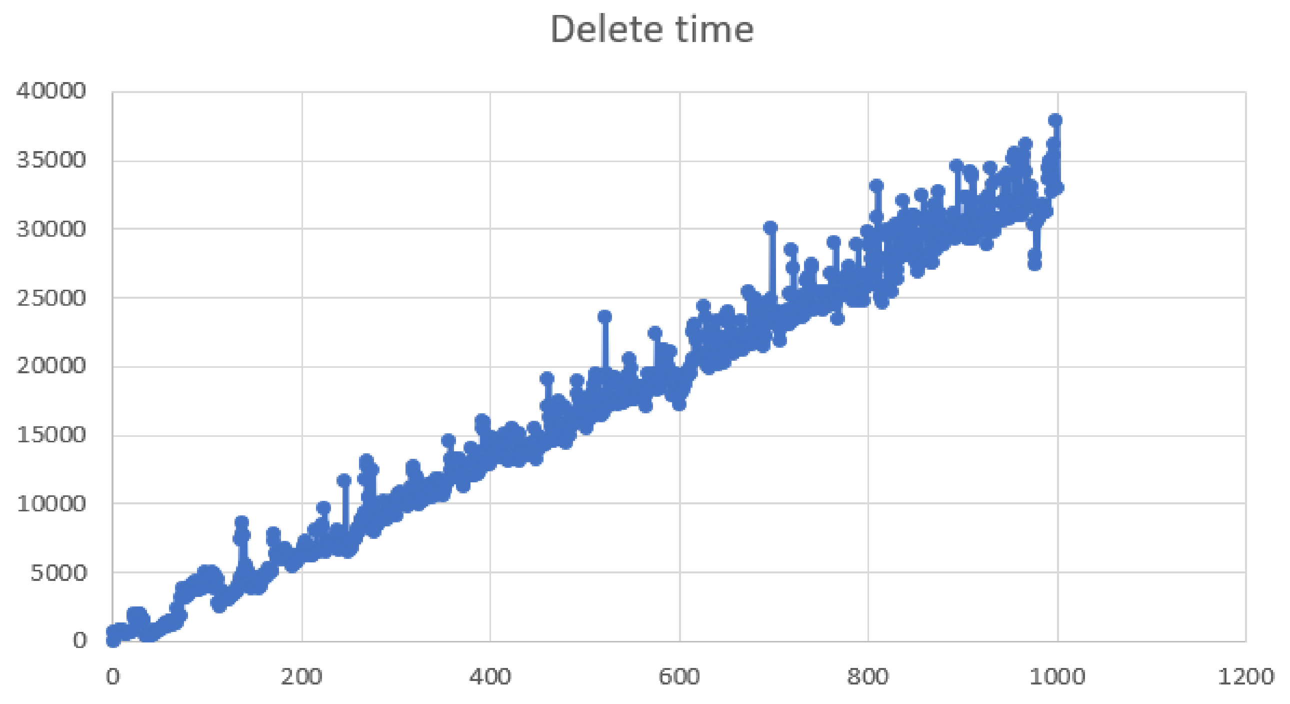
Use the internal key to delete, so time complexity O(1)



Temporal complexity: O(1)

- Delete from 1 million items
- Record 10000 points from whole procedure
- Distance between each point is 1000
- ums

Hashmap_Delete



Temporal complexity: O(1)

- Delete from 1 million items
- Record 1000 points from whole procedure
- Distance between each point is 100
- ums

THANK YOU

Sang Haotian
16722022