

# 【DS】 Day16

☰ Tags	
📅 Date	@June 10, 2022
☰ Summary	Symbol Table Elementary Implementation and Binary Search

## 【Week 4】 Symbol Table

### 4.5 Elementary Implementation

Data structure: Maintain an (unordered) linked list of key-value pairs

Search: Scan through all keys until find a match

Insert: Scan through all keys until find a match; if no match add to front.

#### *Binary search*

```
public Value get(Key key) {
    if (isEmpty()) return null;
    int i = rank(key);
    if (i < N && keys[i].compareTo(key) == 0) return vals[i];
    return null;
}
```

```
private int rank(Key key) {
    int lo = 0, hi = N - 1;
    while (lo <= hi) {
        int mid = lo + (hi - lo) / 2;
        int cmp = keys[mid].compareTo(key);
        if (cmp > 0) hi = mid - 1;
        else if (cmp < 0) lo = mid + 1;
        else return mid;
    }
    return lo;
}
```

## 4.6 Ordered Operations

<code>public class ST&lt;Key</code>	<code>extends Comparable&lt;Key&gt;, Value&gt;</code>	
<code>ST()</code>		<i>create an ordered symbol table</i>
<code>void put(Key key, Value val)</code>		<i>put key-value pair into the table (remove key from table if value is null)</i>
<code>Value get(Key key)</code>		<i>value paired with key (null if key is absent)</i>
<code>void delete(Key key)</code>		<i>remove key (and its value) from table</i>
<code>boolean contains(Key key)</code>		<i>is there a value paired with key?</i>
<code>boolean isEmpty()</code>		<i>is the table empty?</i>
<code>int size()</code>		<i>number of key-value pairs</i>
<code>Key min()</code>		<i>smallest key</i>
<code>Key max()</code>		<i>largest key</i>
<code>Key floor(Key key)</code>		<i>largest key less than or equal to key</i>
<code>Key ceiling(Key key)</code>		<i>smallest key greater than or equal to key</i>
<code>int rank(Key key)</code>		<i>number of keys less than key</i>
<code>Key select(int k)</code>		<i>key of rank k</i>
<code>void deleteMin()</code>		<i>delete smallest key</i>
<code>void deleteMax()</code>		<i>delete largest key</i>
<code>int size(Key lo, Key hi)</code>		<i>number of keys in [lo..hi]</i>
<code>Iterable&lt;Key&gt; keys(Key lo, Key hi)</code>		<i>keys in [lo..hi], in sorted order</i>
<code>Iterable&lt;Key&gt; keys()</code>		<i>all keys in the table, in sorted order</i>