

## Week1 总结

0. 有同学问 git 相关的问题。git 还是建议大家好好学习下，工作必备技能。

推荐一个 git 操作相关的小游戏：<https://learngitbranching.js.org/>

通关后，工作中 git 相关的问题基本上难不倒大家了。

当然，了解如何提交到远端仓库，应该就能满足提交作业的需求，不是说要通关游戏后才开始提交作业[Onlooker]

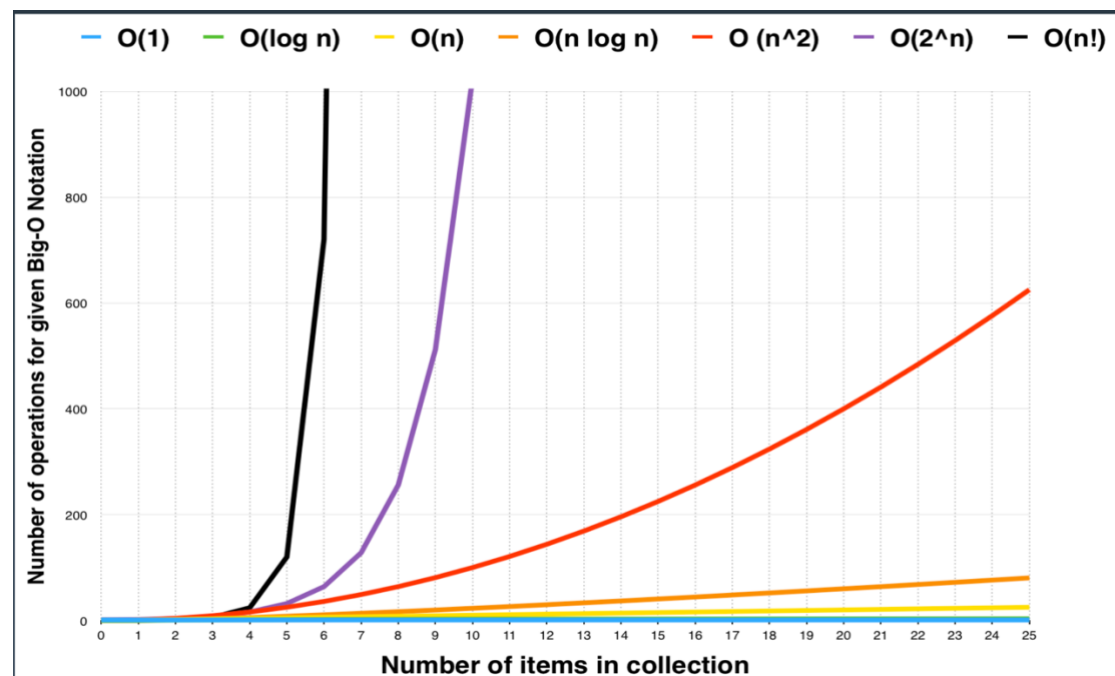
### 1. 作业：知识脑图

切题四件套：clarification, possible solutions (compare time/space, optimal), repeat coding, test cases

### 五遍刷题

### 2. 时间复杂度

```
O(log(n))  for (int i = 1; i < n; i = i * 2) {  
             System.out.println("Hey - I'm busy looking at: " + i);  
           }  
  
O(k^n)     int fib(int n) {  
             if (n < 2) return n;  
             return fib(n - 1) + fib(n - 2);  
           }
```



- 1) 二叉树遍历：前中后  $O(n)$
- 2) 图的遍历： $O(n)$
- 3) dfs, bfs:  $O(n)$
- 4) 二分查找： $O(\log n)$

### 3. array

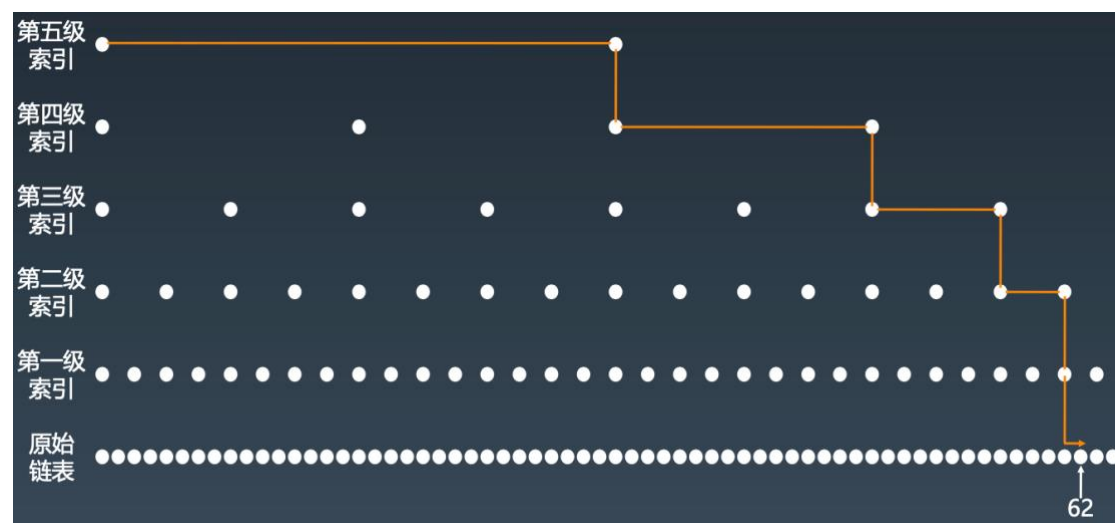
- 1) 增加元素，中间插入要移动后面的  $O(n)$
- 2) 删除元素，中间删除要把后面的前移  $O(n)$

### 4. linked list

- 1) 复杂度：  
Prepend  $O(1)$  Append  $O(1)$  insert  $O(1)$  delete  $O(1)$   
Lookup  $O(n)$

### 5. skip list

- 1) 只能用于链表里的元素有序的情况
- 2) 对标替代平衡树和二分查找
- 3) 插入 删除 搜索 都是  $\log n$
- 4) 空间复杂度  $O(n)$



$n/2$ 、 $n/4$ 、 $n/8$ 、第  $k$  级索引结点的个数就是  $n/(2^k)$

假设索引有  $h$  级，最高级的索引有 2 个结点。  $n/(2^h) = 2$ ，从而求得  $h = \log_2(n) - 1$

### 6.

Stack: 先入后出；添加，删除皆为  $O(1)$

Queue: 先入先出；添加，删除皆为  $O(1)$

查询都为  $O(n)$

PriorityQueue: 插入  $O(1)$ ，取出  $O(\log n)$

-按照元素的优先级取出

-底层实现多样+复杂: heap, bst, treap

-作业: 分析 pq source code

## **Week1 作业**

### **1. 用 add first 或 add last 改写 deque 的课堂代码**

详见 week1\_hw\_1.java

### **2. Java priority queue source code 分析:**

详见 week1\_hw\_2.java

3. 设计循环队列 详见 week1\_hw\_3\_4.java

4. 接雨水 详见 week1\_hw\_3\_4.java