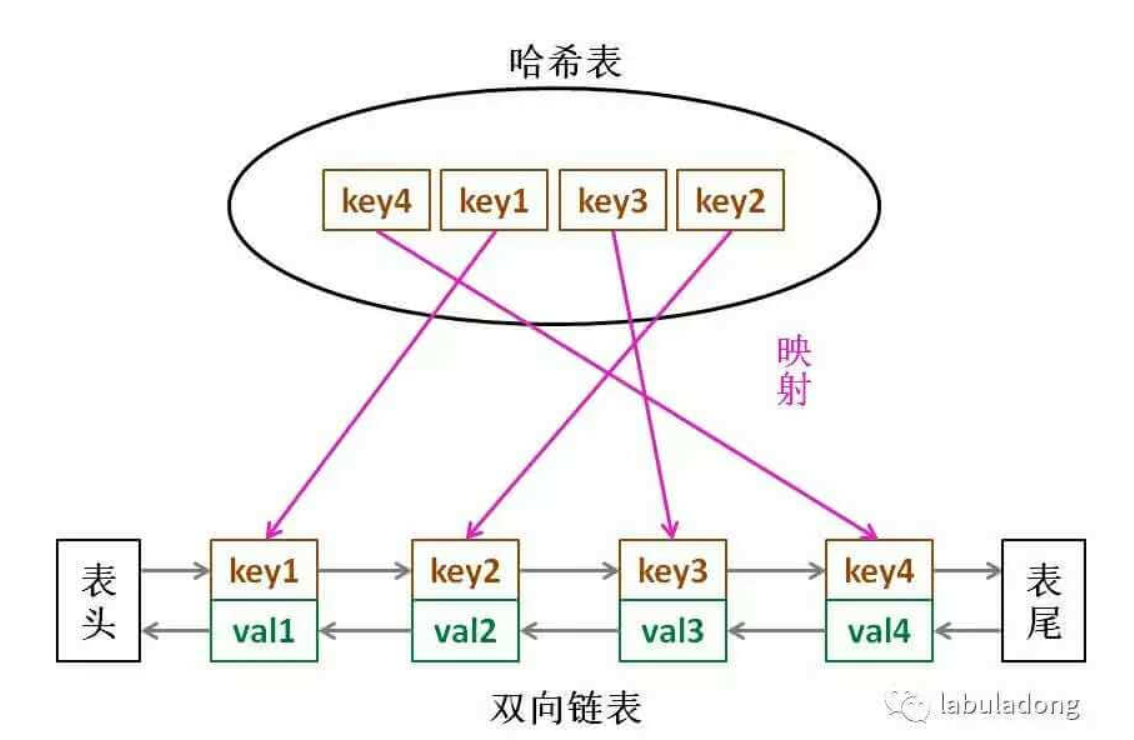
put 和 get 方法的时间复杂度为 O(1)，我们可以总结出 cache 这个数据结构必要的条件：查找快，插入快，删除快，有顺序之分。

Get用hash， put用链表（登记hash）



什么是**服务注册与服务发现**？

**服务注册**，就是将提供某个服务的模块信息(通常是这个服务的ip和端口)注册到1个公共的组件上去（比如: zookeeper\consul）。

**服务发现**，就是新注册的这个服务模块能够及时的被其他调用者发现。不管是服务新增和服务删减都能实现自动发现。

你可以理解为：

//服务注册

NameServer->register(newServer);

//服务发现

NameServer->getAllServer();

应用场景：**各个微服务相互独立，每个微服务，由多台机器或者单机器不同的实例组成，各个微服务之间错综复杂的相互关联调用**。

任何情况下，处理器检测到event发生，通过异常表（exception table）跳转到专门处理这类事件的操作系统子程序（exception handler）。

异步异常由事件产生，同步异常是执行一条指令的直接产物。  
类别包含**中断（异步）**，**陷阱（同步）**，**故障（同步）**，**终止（同步）**。

* 中断——异步发生，处理器IO设备信号的结果。
* 陷阱——有意的异常。最重要的用途是在用户程序和内核之间提供一个像过程一样的接口，叫做系统调用。
* 故障——潜在可恢复的错误造成的结果。如果能被修复，则重新执行引起故障的指令，否则终止。
* 终止——不可恢复的致命错误造成的结果。

编译器并不是把函数模板处理成能够处理任意类的函数；编译器从函数模板通过具体类型产生不同的函数；编译器会对函数模板进行两次编译：在声明的地方对模板代码本身进行编译，在调用的地方对参数替换后的代码进行编译。

