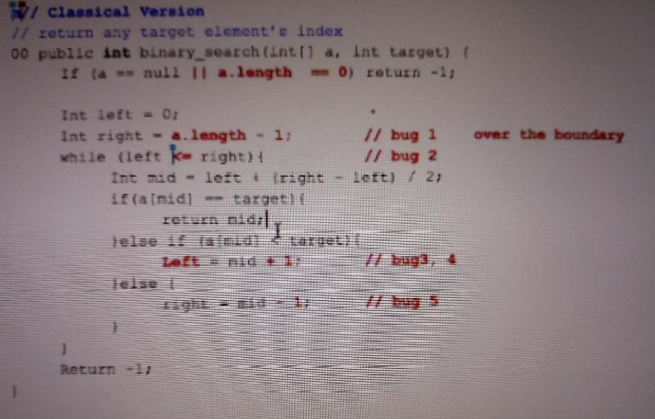
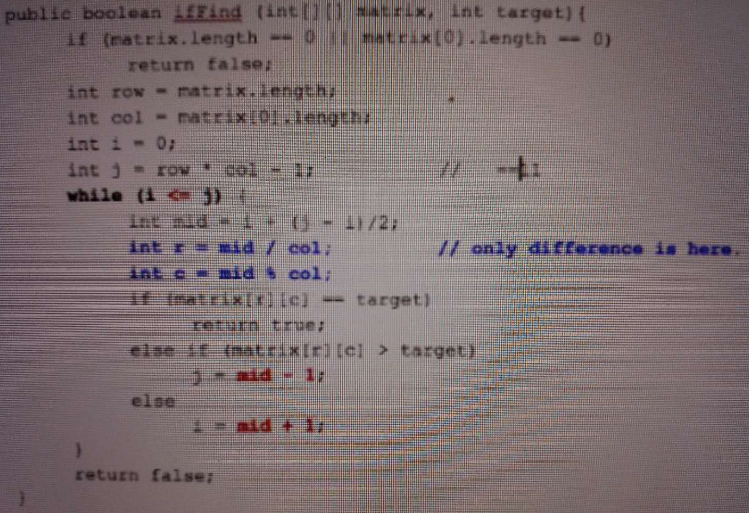


**经典二分法**：

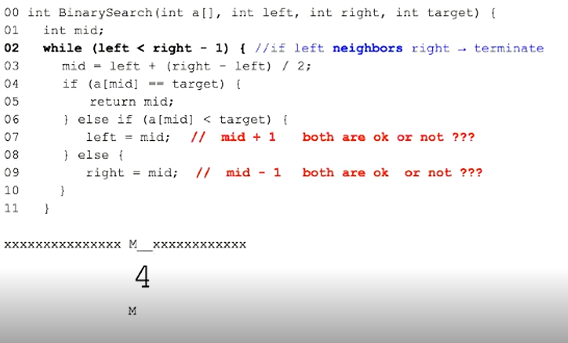


**二维Binary Search 🡪 变种一**

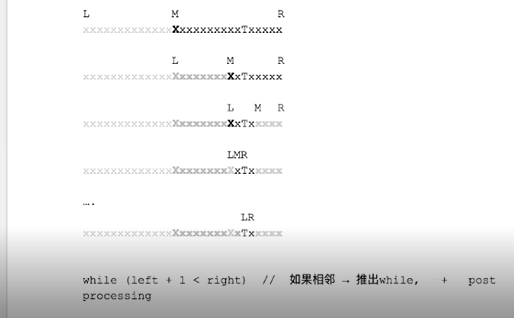


**找相邻最近的值：🡪 变种二**





我们找的target，必定在Left 和 Right之间的两个之一。🡪 **重新看1:43:30** 🡪 如果是mid+1就淘汰掉正确答案了

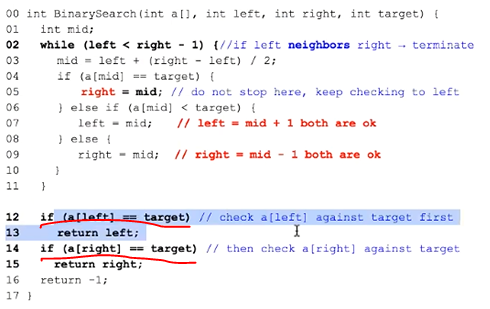


**如果相邻，就退出循环**！！

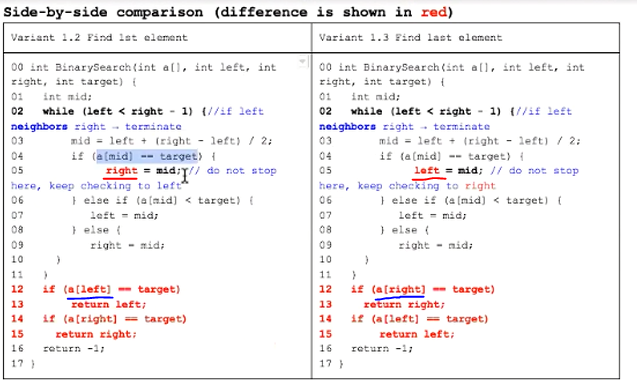
**🡪 变种四，找第一出现的位置。**

**（不知道变种三在哪里，按视频顺序）**



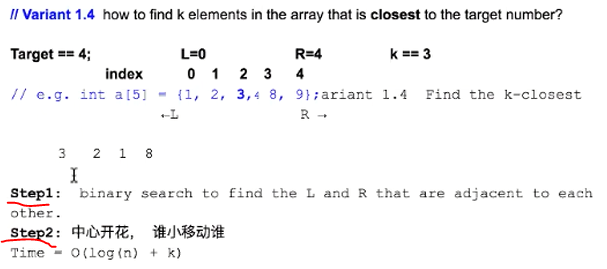


**🡪 变种五，找最后出现的位置。**

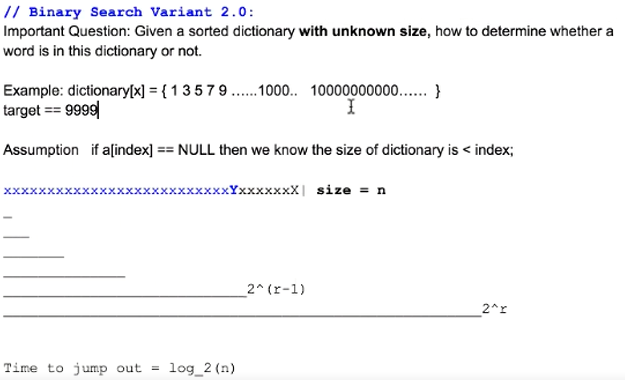


注意一下，为何是先检查left，或者right？

**🡪 变种六，找最近的K个数字。**



**🡪 变种七，找最近的K个数字。**



假如十倍跳，二倍跳:



这里在讨论是十倍跳，还是二倍跳

