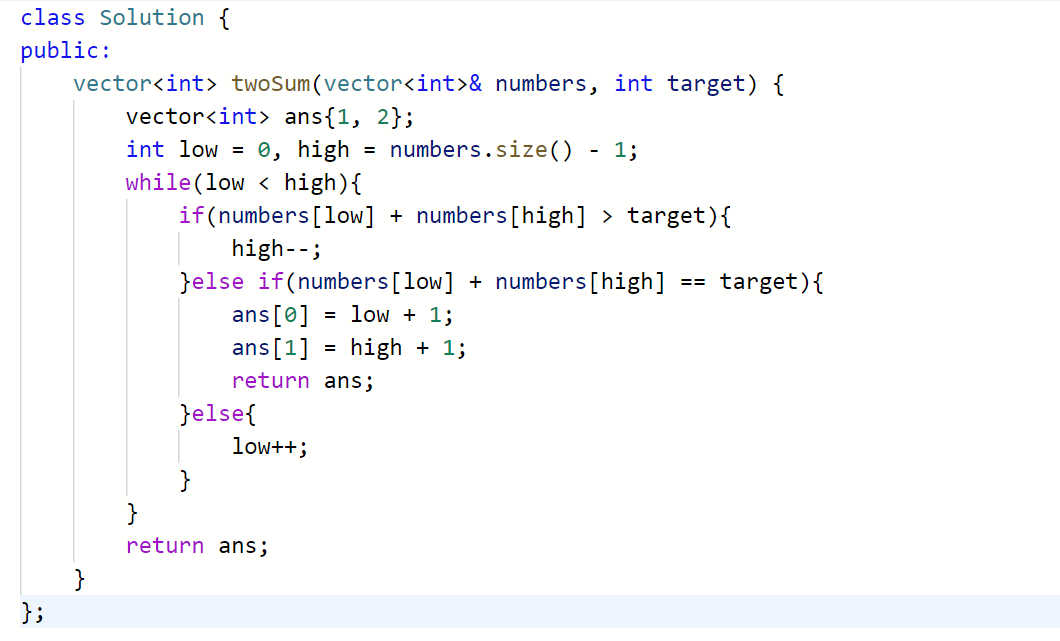
首先贴上代码



关于这题的双指针，首先把数据的表示画出来，也就是一个数轴，low在最左边，high在最右边，这个题的关键在于题目的特性的应用，numbers是一个升序的数组，必须要利用到这点才能写出时间复杂度最低的算法。

关注点来到这里，如果low和high加起来比target小，即（low + high）< target，可知low + （high + posNum）> target，即high后面的任何数和low加起来都要大于target，所以可以将high往前挪，low往后挪同理，然后循环到low + high = target，得到结果

有一种想法就是low和high移动的过程中其实记录了一种信息，就是承载了其作为有效数据与无效数据的分界线。

\*\*\*\*\*\*\*\*为什么不把low前面的数和high前面的数加起来？

----从最边缘遍历过来了，已经证明了他们结合起来不会等于target。

如果左指针先到达下标 i的位置，此时右指针还在下标 j的右侧，，因此一定是右指针左移，左指针不可能移到 i 的右侧。

如果右指针先到达下标 j 的位置，此时左指针还在下标 i 的左侧因此一定是左指针右移，右指针不可能移到 j 的左侧。

由此可见，在整个移动过程中，左指针不可能移到 i 的右侧，右指针不可能移到 j 的左侧，因此不会把可能的解过滤掉。由于题目确保有唯一的答案，因此使用双指针一定可以找到答案。