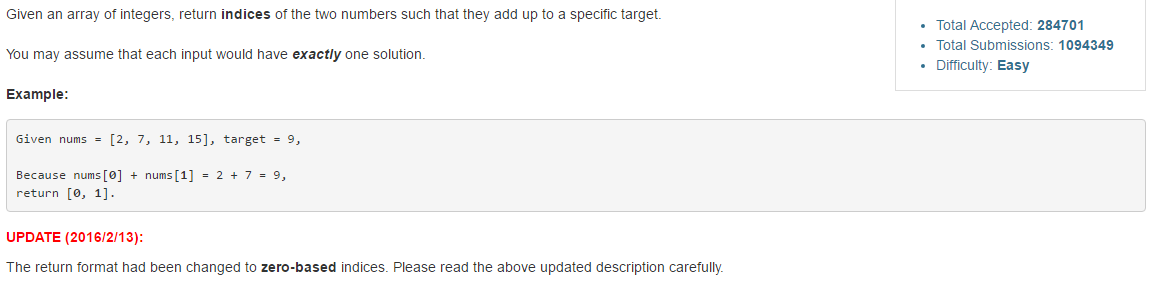
题目：

给定数组A与整数B，要求从数组A中找出第一次满足和为B的两个元素，返回其位置。

例如A=[2,7,11,15],B=9,返回值为[0,1].



思路：

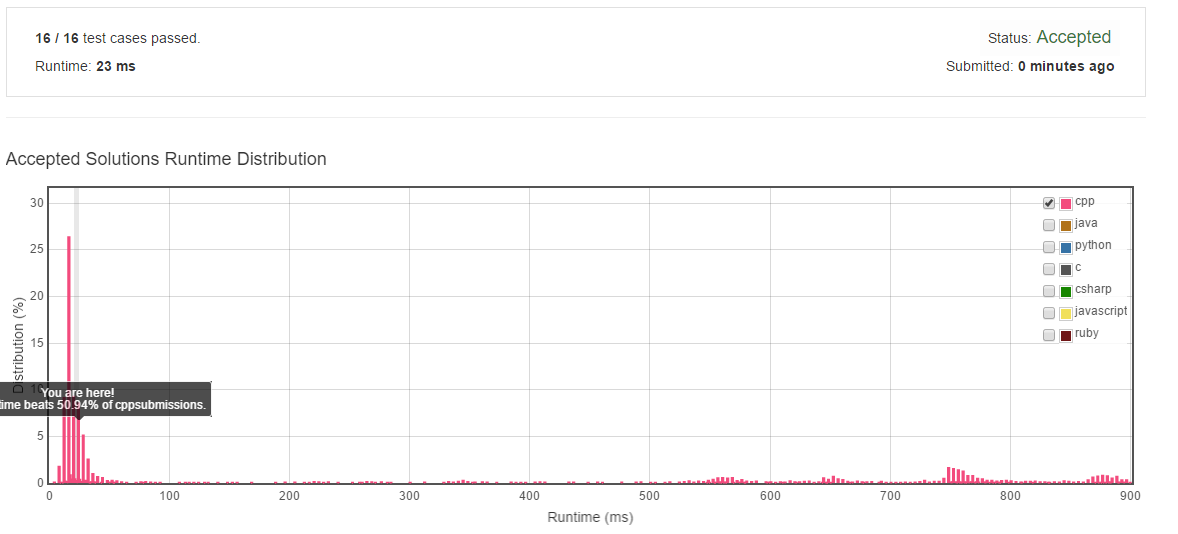
首先开始是想先将A排序，以减少不必要的比较。但后面发现排序之后就改变了A元素的位置，必然出现错误结果。

后在网上看到另一思路：是将A所有的元素放入一个散列表T中。在遍历数组A以构建该表T时，假如当前预计存入表中的A的元素为A[i]，则首先在T中查询是否存在该元素，如果不存在则将其插入T中，并同时在T中寻找是否存在B-A[i]，如果存在则代表找到符合条件的元素，否则直到A遍历完成。算法将逐个比较转变为单个查找，并且可以采用查找效率最高的哈希表作为散列表，有效提高算法效率。

解决代码：



运行结果：



缺陷：

尽管代码完成了网上的测试，但本身存在缺陷。例如，对于A=[3,2,3],B=6这种情形就无法返回正确结果。因为上述解决方案要求数组A中的元素都是唯一的。