二分查找模版

```cpp
class Solution {
public:
    int searchInsert(vector<int>& nums, int target) {
        int N = nums.size();
        int left = 0;
        int right = N - 1;
        int result = 0;
        while (left ⩽ right) {
            // 计算中间索引
            int mid = (left + right) / 2;
            if (target < nums[mid]) {
                // 这时候结果一定在[left,mid)
                right = mid - 1;

                // 因为强制类型转换是向下取整的，因此如果left+right是奇数
                // 那么代码中的mid是要小于(left + right) / 2的
                // 因此，插入位置就可能是mid了。
                result = mid;
            } else if (target == nums[mid]) {
                result = mid;
                break;
            } else {
                // 这时候结果一定在（mid,right】，想想为什么mid为开区间，很重要
                // 不然无法理解result=mid+1;
                left = mid + 1;   // 更新 left
                result = mid + 1; // 因为 target 大于 mid，更新结果为 mid + 1
            }
        }
        return result; // 返回最后的结果
    }
```