二分查找

- 明确变量的含义
- 循环不变量。即target永远在[l...r]中，我们需要维护的就是[l...r]这个数组（通过控制边界来达到目的）。

```
//二分查找
int binarySearch(T arr[], int n, T target){
  int l = 0; r = n - 1; //在 [l ... r]的范围里寻找target
  while( l <= r ){ //区间 [l ... r]依然有效
    //int mid = (l+r)/2; // l+r的和可能会溢出
    int mid = l + (r - l)/2;
    if( arr[mid] == target )
      return mid;
    if(target > arr[mid])//排除mid, target在右侧，更新左边界
      l = mid + 1;
    else
      r = mid - 1;
  }
  return -1;
}
```

```
// 二分查找法, 在有序数组arr中, 查找target
// 如果找到target, 返回第一个target相应的索引index.
// 如果没有找到target, 返回比target小的最大值相应的索引, 如果这个最大值有多个, 返回最大索引
// 如果这个target比整个数组的最小元素值还要小, 则不存在这个target的floor值, 返回-1
int floor(int[] arr, int target){
  int l = -1, r = arr.length -1;
  while( l < r ){
    // 使用向上取整避免死循环
    int mid = l + (r-l+1)/2;
    if( target <= arr[mid])
      r = mid - 1;
    else
      l = mid;
  }
  // 此时 l == r.如果该索引+1就是target本身, 该索引+1即为返回值
  if( l + 1 < arr.length && arr[l+1] == target )
      return l + 1;

  return l;
}
```

```cpp
// 二分查找法, 在有序数组arr中, 查找target
// 如果找到target, 返回最后一个target相应的索引index
// 如果没有找到target, 返回比target大的最小值相应的索引, 如果这个最小值有多个, 返回最小的索引
// 如果这个target比整个数组的最大元素值还要大, 则不存在这个target的ceil值, 返回整个数组元素个数n
int ceil(T arr[], int n, T target){
    assert( n >= 0 );
    // 寻找比target大的最小索引值
    int l = 0, r = n;
    while( l < r ){
        // 使用普通的向下取整即可避免死循环
        int mid = l + (r-l)/2;
        if( arr[mid] <= target )
            l = mid + 1;
        else // arr[mid] > target
            r = mid;
    }
    assert( l == r );
    // 如果该索引-1就是target本身, 该索引+1即为返回值
    if( r - 1 >= 0 && arr[r-1] == target )
        return r-1;
    // 否则, 该索引即为返回值
    return r;
}
```