

2.11

因为输入的序列单调递增，并且如果中间位置的数比位置序数小，那么它之前的数一定都比对应序数小，反之同理，所以采用二分查找的方法。设数据规模为 n ，每循环一次数据规模减半，所以主题函数部分时间复杂度为 $O(\log n)$ （程序因为读入需要 $O(n)$ 的时间，所以时间复杂的为 $O(n)$ ）

2.12

给所有数据加负号，求出对应的最大连续和，在变号之后就能得到对应的最小连续和。主题函数扫描整个数组，所以时间复杂度为 $O(n)$