**学习总结**

姓名\_董金伟 专业\_\_计算机科学与技术\_\_\_ 班级\_\_2班\_ 日期\_2021/12/30\_

1. 当天所学习主要内容（做一总结）

二分查找：

核心代码

|  |
| --- |
| **do** {  mid = (left + right) / 2;  **if** (findNum == arr[mid]) {  flag = **true**;  **break**;  } **else** **if** (findNum > arr[mid]) {  left = mid + 1;  } **else** {  right = mid - 1;  }  } **while** (left <= right); |

核心思想：

用想查找的数与数组中下标最中间的数做比较，如果大于数组中间的数，就可以排除掉数组左边的数。否则就可以排除数组右边的数，循环几次后，就可以查找出想要找的数。

求最大子串的和：

核心代码：

|  |
| --- |
| **for** (i = 0, begin = 0, value = 0; i < arr.length; i++) {  end = i;  value = value + arr[i];  **if** (value > max) {  maxBegin = begin;  maxEnd = end;  max = value;  }  **if** (value < 0) {  begin = i + 1;  value = 0;  }  } |

核心思想：

循环遍历数组，将所有的和加起来，如果这个值大于max的话，就更新开始与结束的下标，如果小于0的话，就将开始的下标写为end的下标+1。

1. 学习中所存在的问题及解决办法

问题：

1. 使用二分查找时，程序值循环一次，无法判断值？

解决：

1. 因为，我的想法是left > right时不执行程序，但是条件中需要写成left < right。改完条件之后，还是无法正确查找出值。仔细观察后发现，当left = right时，这个值可能是查找到的值，所以要将条件改为left <= right。

3．学习收获及建议

学会了二分查找和计算最大子串的和。

二分查找可以快速找到数组中想要的数，计算效率特别高，因为每次可以排除到一般的没有用的数。

单纯循环求最大子串时，首先假如数组中全是整数，所以最大子串的和应该就是数组中所有数字的和，但是数组中有负数时，就需要判断加上负数的最大值，还是不是最大值，如果不是，最大值小于0之后，就需要将开始的下标移动到当前位置+1的位置。