1. **堆和二叉排序树的区别**

堆只能反映树中父子关系的序列关系，能确保根序列优先级大于左右子树，而二叉排序树反映了左子树根和右子树之间的序列关系，能确保左子树优先级大于根大于右子树。

1. **若只想得到一个序列中第 k (k ≥ 5) 个最小元素之前的部分排序序列，则最好采用什么排序方法?**

最好采用堆排序算法将序列储存最小堆中，进行k次排序就能输出前k个数的排序了

1. **已知由𝑛 (𝑛 ≥ 2) 个正整数构成的集合𝐴 = {𝑎: 0 ≤ 𝑘 < 𝑛}，将其划分为两个不相交的子集𝐴1和𝐴2，元素个数分别是𝑛1和𝑛2，𝐴1和𝐴2中的元素之和分别为𝑆1和𝑆2。设计一个尽可能高效的划分算法，满足|𝑛1 − 𝑛2|最小且|𝑆1 − 𝑆2|最大。**

可以利用快排中的思想，对A集合进行快排，并对每一个排好的数位置进行判断，如果这个数的位置是中间位置，则成功将A集合分为两个题目要求集合。

1. **以上实验能否说明：就平均性能而言，BST的查找与折半查找差不多，为什么？**

不能，根据以上的实验数据只能说明折半查找性能更高，因为折半查找需要一个已经排好序的序列，而BST使用的是随机序列，如果BST是一棵平衡树，则与折半查找性能差不多。

1. **平均时间空间复杂度：**

折半查找：时间复杂度：O(logn)

BST：时间复杂度：最好：O(logn)，最坏：O(n)；空间复杂度：O(n)

堆排序：时间复杂度：O(nlogn)；空间复杂度：O(n)