1. 1. 简单选择排序

2. 直接插入排序、冒泡排序

3. 直接插入排序

4. 堆排序

2. 41, 37, 17, 25, 54, 68, 58, 93, 76

25, 37, 17, 41, 54, 58, 68, 93, 76

17, 25, 37, 41, 54, 58, 68, 76, 93

17, 25, 37, 41, 54, 58, 68, 76, 93

最大深度：4

递归调用：9次

第二次：41, 37, 17, 25

3. 附加空间复杂度：堆排序O(1)，快速排序O(logn)，归并排序O(n \* logn)

所以考虑空间用堆排序

稳定性：堆排序不稳定，快速排序不稳定，归并排序稳定

所以考虑稳定性归并排序

平均情况时间复杂度：堆排序O(n \* logn),，快速排序O(n \* logn)，归并排序O(n \* logn)

所以考虑时间三种都可以

4. 14, 17, 41, 35, 9, 32, 68, 53, 76, 23

14, 9, 32, 23, 17, 41, 35, 53, 76, 68

9, 14, 17, 23, 32, 35, 41, 53, 68, 76

5. 14, 17, 35, 9, 37, 53, 21, 46, 68

14, 17, 9, 35, 37, 21, 46, 53, 68

14, 9, 17, 35, 21, 37, 46, 53, 68

9, 14, 17, 21, 35, 37, 46, 53, 68

9, 14, 17, 21, 35, 37, 46, 53, 68

9, 14, 17, 21, 35, 37, 46, 53, 68

9, 14, 17, 21, 35, 37, 46, 53, 68

9, 14, 17, 21, 35, 37, 46, 53, 68

9, 14, 17, 21, 35, 37, 46, 53, 68

6. 9, 14, 53, 35, 17, 37, 68, 21, 46

7. 68, 35, 53, 21, 9, 37, 14, 17

68, 53, 35, 37, 21, 9, 17, 14

68, 53, 37, 35, 17, 21, 9, 14

68, 53, 37, 35, 21, 17, 14, 9

68, 53, 37, 35, 21, 14, 17, 9

68, 53, 37, 35, 21, 17, 14, 9

68, 53, 37, 35, 21, 17, 14, 9

8. 617, 464, 314, 335, 344, 253, 237, 231, 121, 176, 19, 46

617, 464, 344, 335, 314, 253, 237, 231, 176, 121, 46, 19

617, 464, 344, 335, 314, 253, 237, 231, 176, 121, 46, 19

9.

void direct\_inserting\_sort() {

for (n = 2; n <= sz; n++) {

pivot = R[n];

R.erase(R.begin() + n);

for (i = 1; i <= n - 1; i++) {

if (R[i] > R[n]) {

R.insert(R.begin() + i, pivot);

}

}

}

}

10.

struct record {

int key;

int value;

record\* next;

};