8.3

（d = 8）60 35 105 25 30 10 28 12 78 100 120 50 85 40 90 15

（d = 4）**30** 10 **28** 12 **60** 35 **90** 15 **78** 40 **105** 25 **85** 100 **120** 50

（d = 2）28 10 30 12 60 15 78 25 85 35 90 40 105 50 120 100

（d = 1）10 12 15 25 28 30 35 40 50 60 78 85 90 100 105 120

8.4

105 50 30 25 85 40 100 12 10 28（原始）

50 30 25 85 40 100 12 10 28 105（1->10）

30 25 50 40 85 12 10 28 100 105（1->3,4->5,6->9）

25 30 40 50 12 10 28 85 100 105（1->2,3->4,5->8）

10 12 25 28 30 40 50 85 100 105

8.5

30 25 40 12 15 35 10 28 **50** 120 85 100 90 60 105 78

25 12 15 10 28 **30** 40 35 **50** 85 100 90 60 105 78 **120**

12 15 10 **25** 28 **30** 35 **40 50** 60 78 **85** 100 90 105 **120**

10 **12** 15 **25** 28 **30** 35 **40 50** **60** 78 **85** 90 **100** 105 **120**

8.6

void **sort**(int A[], int n)

{

    int i, j;

    int temp;

    for (i = 0;i < n;i++)

    {

        for (j = i+1;j <= n; j++)

        {

            if (A[i] % 2 > A[j] % 2)

            {

                temp = A[i];

                A[i] = A[j];

                A[j] = temp;

            }

        }

    }

}

8.7

void **sort**(int A[], int n)

{

    int i, j;

    int temp;

    for (i = 0;i < n;i++)

    {

        for (j = i+1;j <= n; j++)

        {

            if (A[i] % 3 > A[j] % 3)

            {

                temp = A[i];

                A[i] = A[j];

                A[j] = temp;

            }

        }

    }

}

8.8

int **kth**(int s, int e, int k){

    if(s <= e){ *// 终止条件*

        int x = a[s];

        int i = s, j = e;

        while(i < j){

            while(i < j && a[j] >= x) --j;

            while(i < j && a[i] <= x) ++i;

            int t = a[i]; a[i] = a[j]; a[j] = t;

        }

        a[s] = a[i];

        a[i] = x;

        if(i-s+1 == k)

            return a[i];

        else if(i-s+1 < k)

**kth**(i+1,e,k-(i-s+1));

        else

**kth**(s,i-1,k);

    }

}

8.10

大根堆：120

100 105

85 90 60 78

30 25 40 12 15 35 10 28

小根堆：10

12 15

25 28 30 35

78 60 85 90 100 105 120

8.12

堆排序