9.0.

插入排序过程：

Date. /

1. 31 4 1 5 9 2 6 5

1 3

1 3 4

1 1 3 4

1 1 3 4 5

1 1 3 4 5 9

1 1 2 3 4 5 9

1 1 2 3 4 5 6 9

1 1 2 3 4 5 5 6 9

希尔排序

2.

① increment = 1.

9 8 7 6 5 4 3 2 1 → 1 2 3 4 5 6 7 8 9

② increment = 3.

→ 1 2 3 4 5 6 7 8 9

② increment = 7

→ 1 2 3 4 5 6 7 8 9

gap 要从大到小

① increment = 7

9 8 7 6 5 4 3 2 1

→ 2 1 7 6 5 4 3 9 8

② increment = 3

⇒ 2 1 4 3 5 7 6 9 8

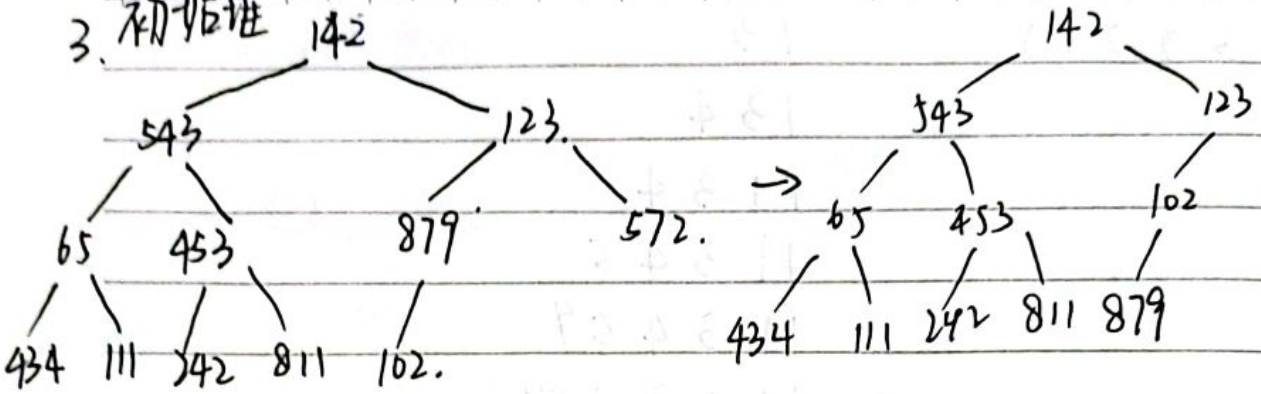
③ increment = 1

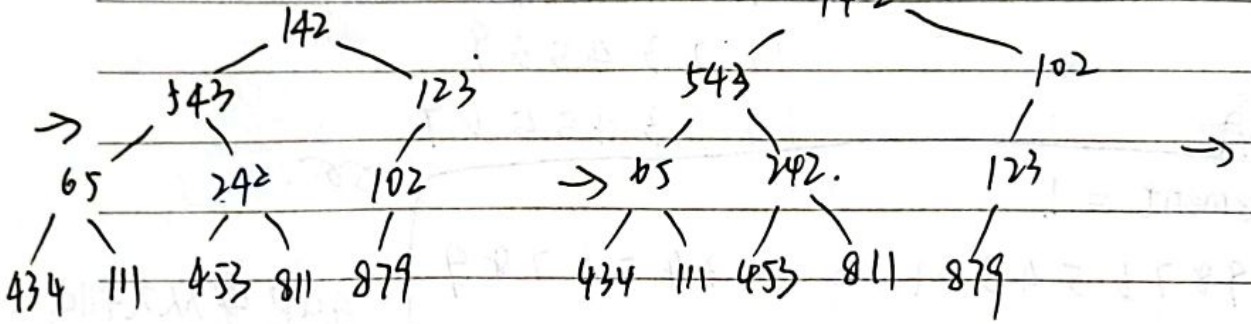
⇒ 1 2 3 4 5 6 7 8 9

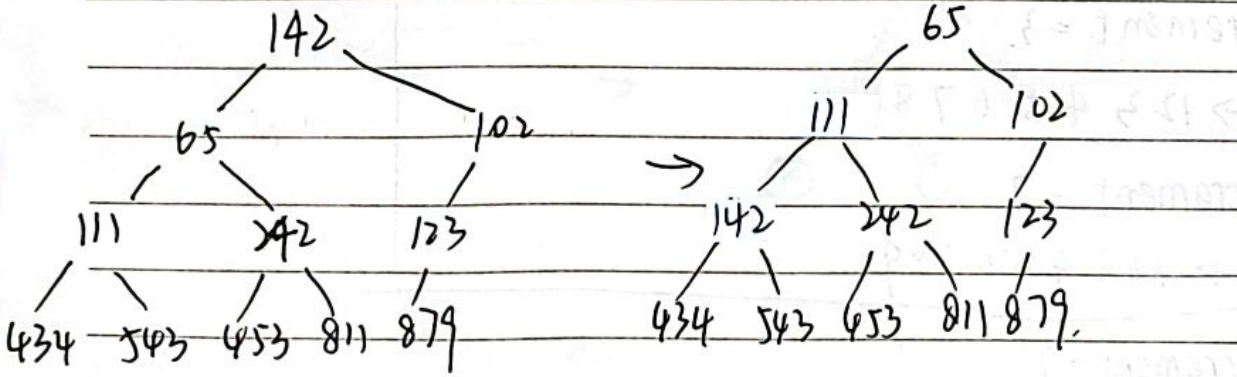
Date. /

1° 调整为最小堆

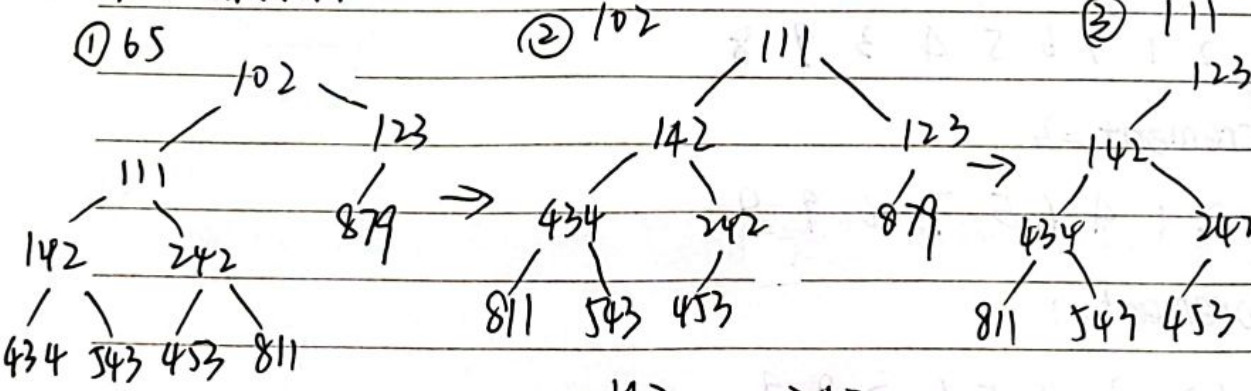
3. 初始堆

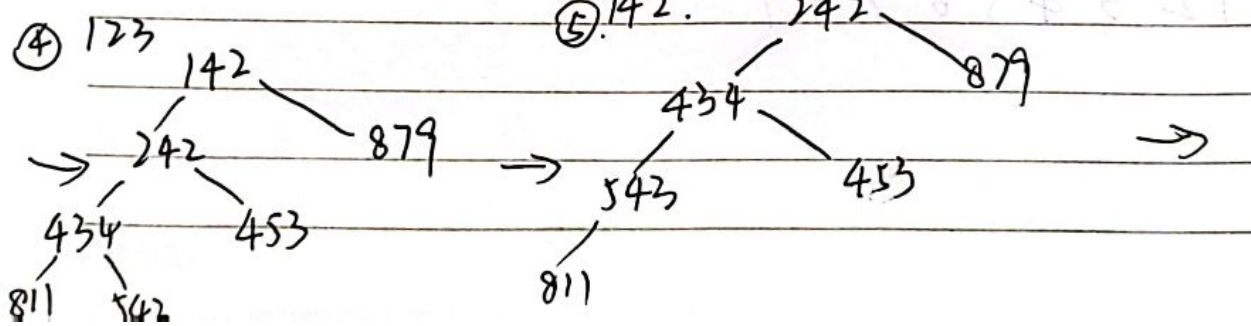


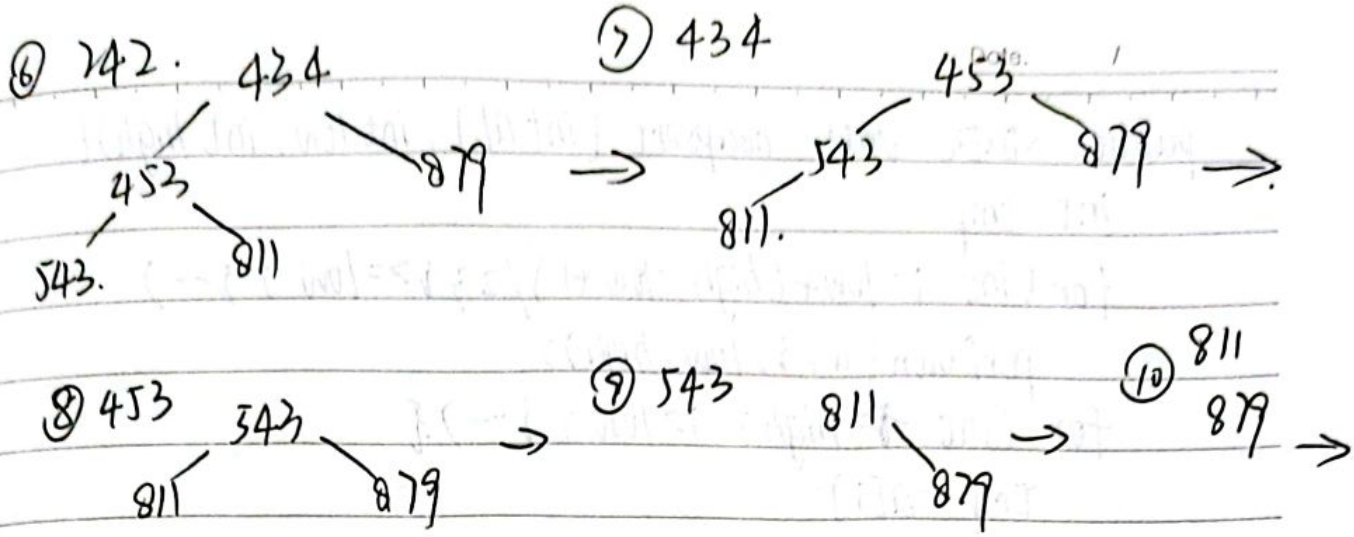




2° 依次输栈顶







故排序结果为 65, 102, 111, 123, 142, 242,
434, 453, 543, 811, 879

?

4.
```
private static int leftchild (int i, int low) {
    return (2*i+1-low);    2*(i-low)+1+low
}
public static void perDown (int[] a, int i, int low, int high)
    int child, tmp;
    for (tmp = a[i]; leftchild (i, low)<high+1; i=child ){
        child = leftchild (i, low);
        if (child != high && a[child] < a[child+1])
            child++;
        if (tmp < a[child]).
            a[i] = a[child];
        else
            break;
    }
    a[i] = tmp;
}
```

Date. /

```
public static void heapsort [int a[], int low, int high){
    int tmp;
    for (int i= low+(high-low+1)/2; i>=low ; i--)
        perDown(a, i, low, high);
    for (int i =high; i>low ; i--){
        tmp =a[i];
        a[i] =a[low];
        a[low] =tmp;
        perDown (a, low, low, i-1);
    }
}
```

5. 归并排序. 3, 1, 4, 1, 5, 9, 2, 6.

2个一组 (13)(14)(59)(26)

4个一组 (1134)(2569)

8个一组 (11234569)