

Homework

Problems on DH

2.1

The reason is that the last action is a little different from the actions in the [loop](#). In other words, the only necessary action in the last step is add the salary pointed at to the noted number, without "pointing to the next salary". So if we perform the loop N times, the program might go wrong, because after moving the pointer for $N-1$ times in the loops, the processor has pointed to the last salary on the list, so the program might go wrong if we instruct the processor to point to the inexistent "next salary" of the last salary on the list in the coming N_{th} loop.

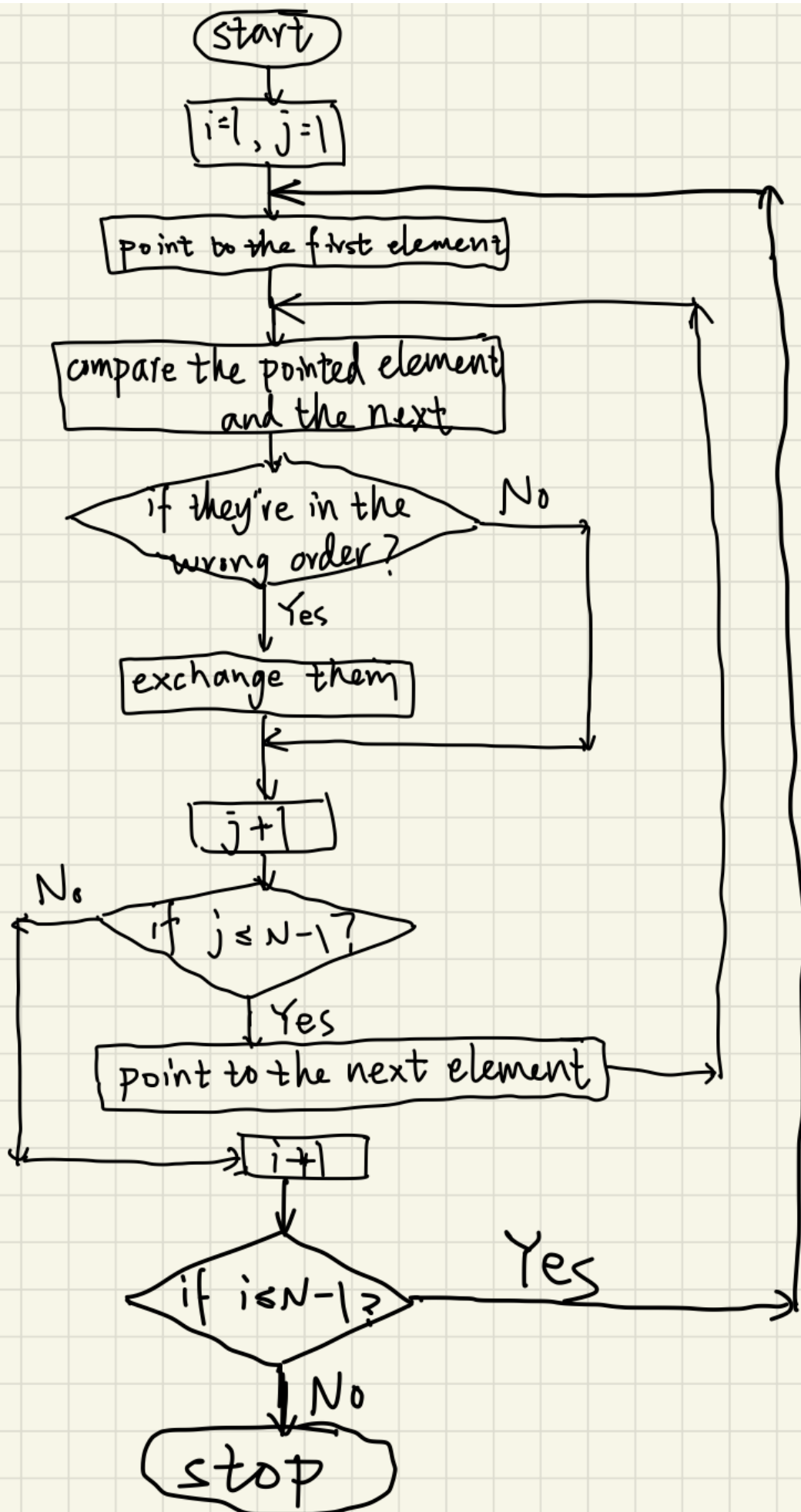
2.2

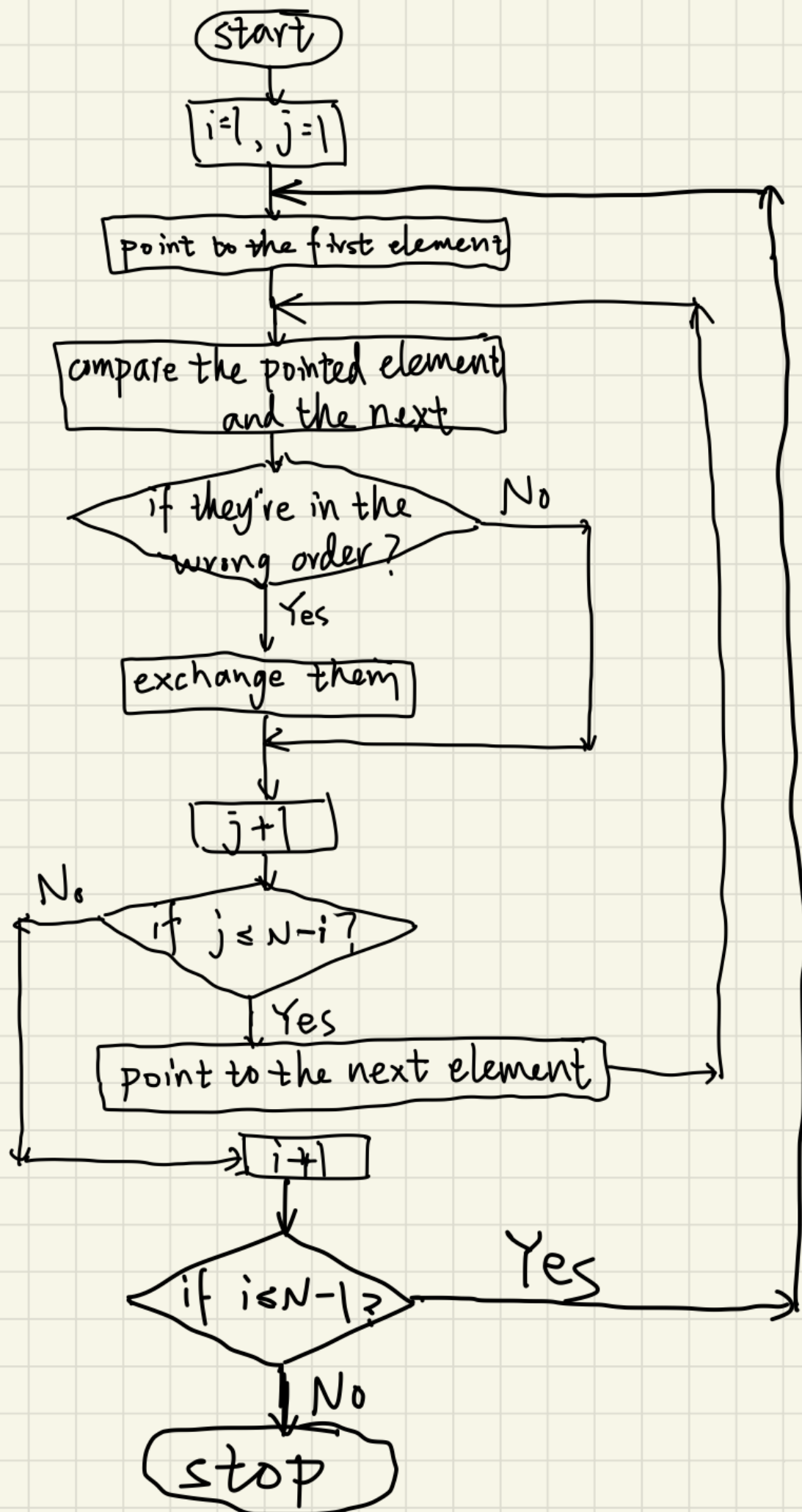
(a) By the $N-1$ outer loops, we will find out the largest, second-largest, ... and the penultimate-smallest elements, then the only remaining element is the very smallest element, and we don't need another loop to find it.

(b)

- (1) let i be 1;
- (2) do the following $N-1$ times:
 - (2.1) point to the first element;
 - (2.2) do the following $N-i$ times:
 - (2.2.1) compare the element pointed to with the next element;
 - (2.2.2) if the compared elements are in the wrong order, exchange them;
 - (2.2.3) point to the next element;
 - (2.3) add 1 to i ;

2.3





2.4

(a)

- (1) let S be \emptyset , P be 1;
- (2) point to the first element;
- (3) do the following $N-1$ times:
 - (3.1) if the pointed element is even, then add the element to S ;
 - (3.2) if the pointed element is odd, then multiply P by the element;
 - (3.3) point to the next element;
- (4) if the pointed element is even, then add the element to S ;
- (5) if the pointed element is odd, then multiply P by the element;
- (6) output S and P ;

(b)

- (1) let S be \emptyset , P be 1;
- (2) point to the first element;
- (3) do the following $N-1$ times:
 - (3.1) if the pointed element is even, then goto(3.3);
 - (3.2) if the pointed element is odd, then goto(3.4);
 - (3.3) add the element to S , then goto(3.5);
 - (3.4) multiply P by the element, then goto(3.5);
 - (3.5) point to the next element;
- (4) if the pointed element is even, then goto(6);
- (5) if the pointed element is odd, then goto(7);
- (6) add the element to S , then goto(8);
- (7) multiply P by the element, then goto(8);
- (8) output S and P ;

2.5

(a)

- (1) do the following while $i < N$:
 - (1.1) do A ;
 - (1.2) add 1 to i ;
- (2) do B ;

(b)

- (1) do the following while $i < N$:
 - (1.1) do A ;
- (2) do the following while $i \geq N$:
 - (2.1) do B ;

(c)

- (1) do A;
- (2) add 1 to i;
- (3) if $i < N$, then goto(1);
- (4) do B;

(d)

- (1) repeat (2)(3)(4) until $i > N$;
- (2) if $i < N$;
- (3) do A;
- (4) add 1 to i;

2.6

Moving 5 from A to C, using B;

move C to B;
move A to C;
move B to A;
move B to C;
move A to C;
move A to B;
move C to B;
move C to A;
move B to A;
move C to B;
move A to C;
move A to B;
move C to B;
move A to C;
move B to A;
move B to C;
move A to C;
move B to A;
move C to B;
move C to A;
move B to A;
move B to C;
move A to C;
move C to B;
move A to B;

move C to B;
move A to C;
move B to A;
move B to C;
move A to C;

2.7

(a)

```
(1) if N=0,then factorial=1;  
(2) let factorial be 1,let i be 1;  
(3) do the following N-1 times:  
    (3.1) multiply factorial by i;  
    (3.2) add 1 to i;  
(4) multiply factorial by i;  
(5) output factorial;
```

(b)

```
factorial(N)  
    (1)if N=0,return 0;  
    (2)call factorial(N-1)*N;
```

2.8

```
A(i)  
    (1) if Q,then do the following:  
        (1.1) do A;  
        (1.2) call A(i-1);
```

Extra Problem

Suppose that among the 11 large boxes, x boxes contain medium boxes, and among the $8x$ medium boxes, y boxes contain small boxes. Since there are 102 empty boxes in total, we have the equation below:

$$8y + (8x - y) + (11 - x) = 102$$

Simplify it into this:

$$x + y = 13$$

The total number of boxes can be put like this:

$$11 + 8x + 8y$$

Substite $x + y = 13$ into the equation above then we can solve the total number of boxes:

Total number of boxes: 115