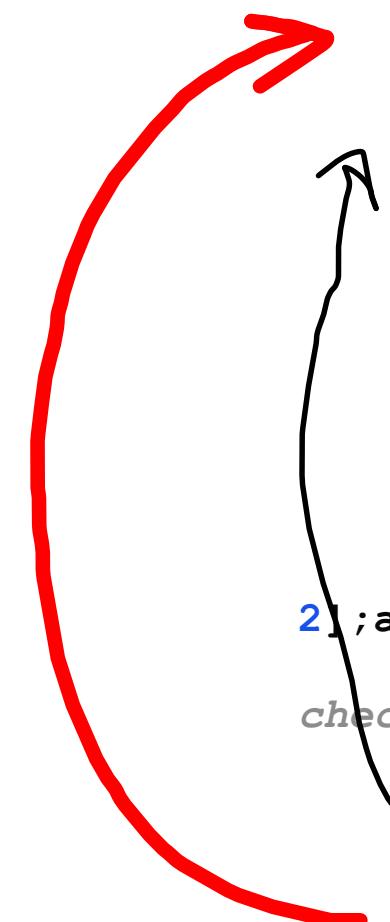


5	15	10	8	
15	5	10	5	
15	10	5	5	
0	1..	2	3	i

PC/2
PC



```

static void heap_sort(int a[])
{
    int temp,i,j,pc; //pc:parent-child
    boolean done;
    for(i=a.length-1;i>0;i--) //last to first
    {
        for(j=0;j<=i;j++)
        {
            done=false;
            pc=j;
            while(pc>0 && pc/2>=0 && done!=true)
            {
                if(a[pc]>a[pc/2])
                    //child parent
                {
                    temp=a[pc]; a[pc]=a[pc/
                    2]; a[pc/2]=temp;
                    pc=pc/2; //go to parent to
                }
                done=false;
            }
            else
            {
                done=true;
            }
        }
        } //j closed
        temp=a[0]; a[0]=a[i]; a[i]=temp; //swap
        largest to last
    }
}

```

Annotations:

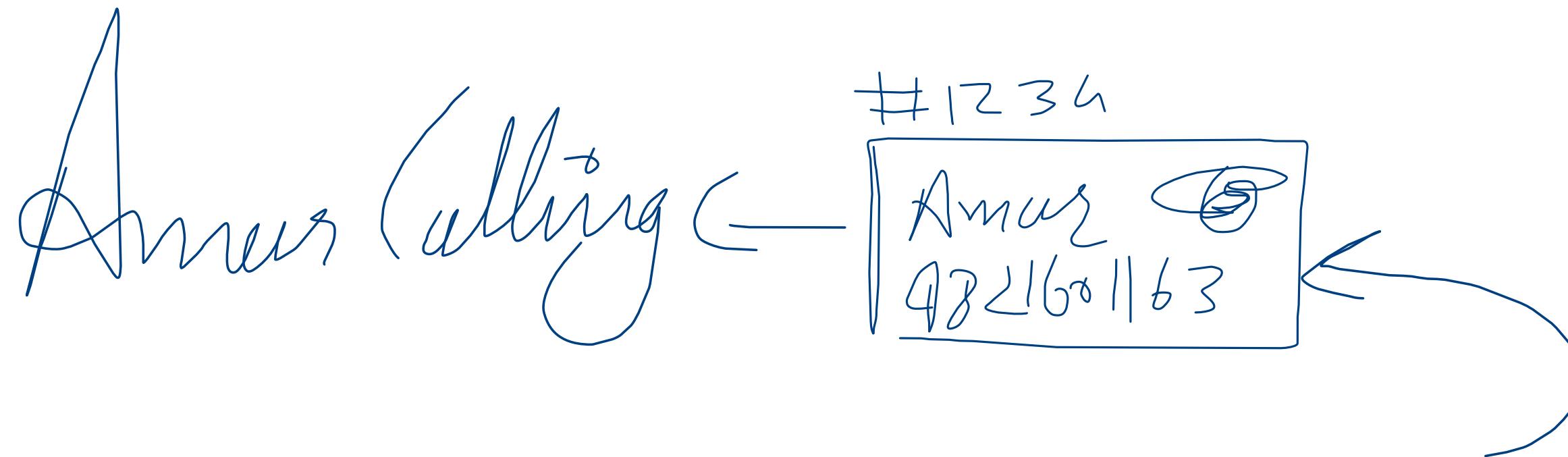
- A red circle highlights the array elements from index 0 to 3.
- The variable *i* is circled in red at index 0.
- The variable *j* is circled in red at index 1.
- The variable *pc* is circled in red at index 2.
- The condition *pc/2 >= 0* is circled in red.
- The swap operation *a[pc] = temp* is circled in red.
- The label *check* is written near the inner loop.
- The label *largest to last* is written at the bottom of the outer loop.

0	15	15	8	19
15	15	10	5	5
15	10	5	5	5
0	1..	2	3	j

Annotations:

- The value 19 is circled in red.
- The values 15, 15, and 10 are circled in green.
- The value 5 is circled in red.
- The variable *j* is circled in red at index 3.
- The variable *pc* is circled in red at index 2.
- The variable *pc/2* is circled in green at index 1.
- The variable *pc* is circled in green at index 0.
- The label *PC/2 PC* is written above the array.
- The label *PC/2 PC* is written below the array.
- The label *PC* is written twice below the array.

Hashing:



Parts:

1 Hash function

CONVERTS DATA TO INDEX FOR STORAGE

2 Hash Table

**SAVES DATA AT INDEX PROVIDED BY HASH FUNCTION
IF ANY COLLISION HANDLE IT**

3 Data

INFO TO STORE FOR LATER TO ACCESS.

Hash Functions:

1. direct hash:

Data is treated as index.

Use only when one-on-one mapping of data and index is possible.

STUDENT 1 has Rollno 1

$$\begin{array}{r} \cancel{X \% 10} \\ \hline 12 \\ 2 \\ \hline 100 \end{array}$$

longer sequential search called clustering before saying yes/no

large memory space used $10 \times 100 \rightarrow 1000$

dynamic but too complex to use

WQPL

