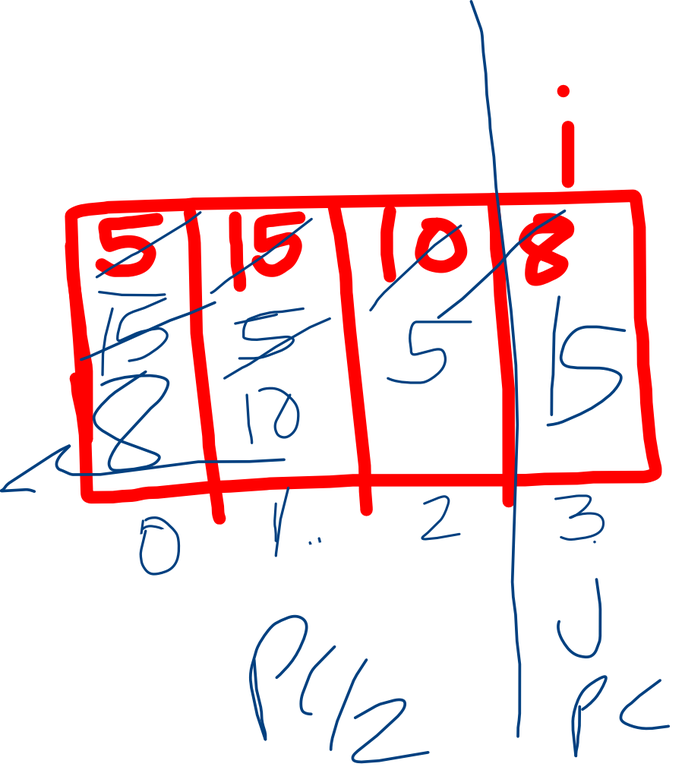
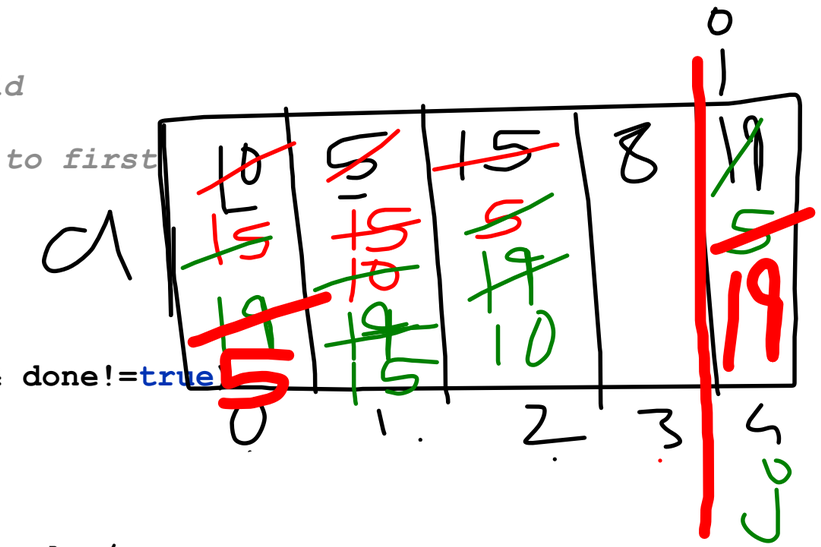
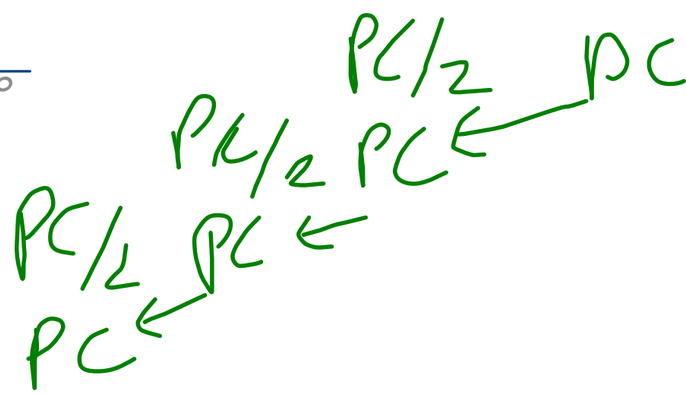
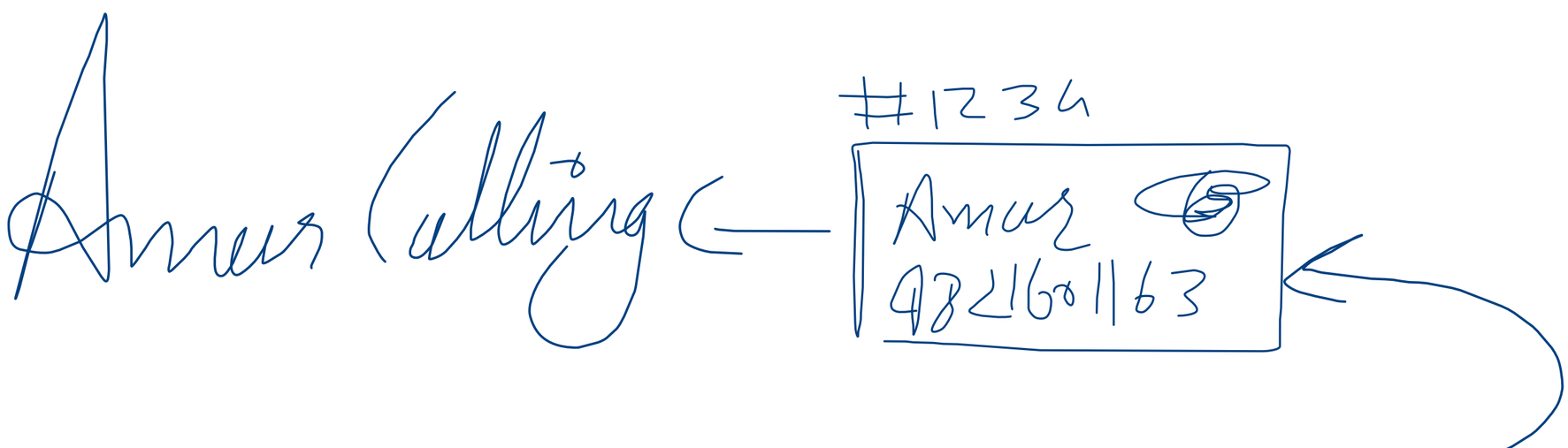



```
static void heap_sort(int a[])
{
    int temp,i,j,pc;//pc:parent-child
    boolean done;
    for(i=a.length-1;i>0;i--)//last to first
    {
        for(j=0;j<=i;j++)
        {
            done=false;
            pc=j;
            while(pc>0 && pc/2>=0 && done!=true)
            {
               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





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

longer sequential search called clustering before saying yes/no

large memory space used 10X100--->1000

dynamic but too complex to use