```
Class Tape{
  Int read();
  Void write(int value);
  Void moveHeadToNext();
  Void moveHeadToBeginning();
  Boolean isTapeReadComplete();
}

5, 17, 11 , 5, 8, true

Int computeSize( Take input ) {
        input.moveHeadToBeginning();
        Int count  = 0;

        while(input.isTapeReadComplete() == false ) {
                Count ++;
                input.moveHeadToNext();
        }
        Return count;
}

Void mergePartitions( input, low, mid, high) {
}

Void sortPartition(Take input, int low, int high ) {
        while(low < high) {
                Int mid = (low + high) / 2;
                sortPartition(input, low, mid);
                sortPartition(input, mid+1, high);
                mergePartitions(input, low, mid, high);
        }
}

//limitation, RAM, so you can read only k values
Tape sort(Tape input, Tape.., int k){
        Int totalSize = computeSize(input);
        Int numPartitions = totalSize / k + 1;

        For( int i = 0; i < numPartitions; i++ ) {
                Int low = i*k;
                Int high = i*k + k;
                if(high > totalSize) {
                        High = totalSize;
                }
```

```
            sortPartition(input, low, high );
        }
        For( int i = 0; i < num; i++ ) {
            mergePartitions(input, low, i * k, i* k + k );
        }
}
```

FR :
Upload pic (5 mb size)
View thumbnails as list sorted by last uploaded.
On click of thumbnail view the images.

NFR:
 5 million active users
Each user 1000 pics
Always available

Storage Space:
5 * 10^6 * 10^3 * 5MB

25* 10^9 MB = 25 PB

```
System -> Read <- Database
       -> Write -> Database
```

 Photo : PhotoId, PhotoPath, Timestamp, User
 User   : USerId, Email, PhoneNo,