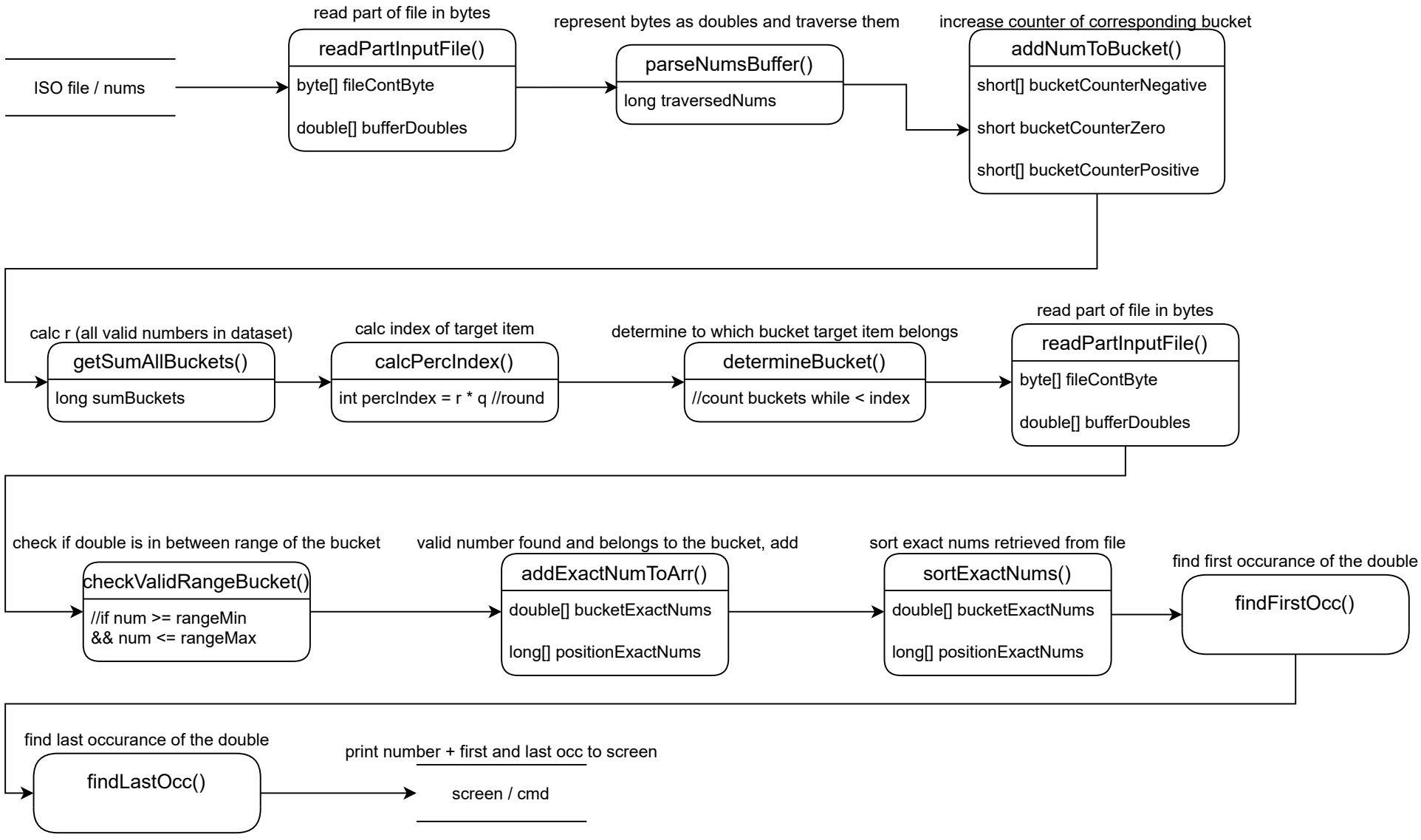
ISO file / nums
read part of file in bytes
readPartInputFile()
byte[] fileContByte
double[] bufferDoubles
represent bytes as doubles and traverse them
parseNumsBuffer()
long traversedNums
increase counter of corresponding bucket
addNumToBucket()
short[] bucketCounterNegative
short bucketCounterZero
short[] bucketCounterPositive
calc r (all valid numbers in dataset)
getSumAllBuckets()
long sumBuckets
calc index of target item
calcPercIndex()
int percIndex = r * q //round
determine to which bucket target item belongs
determineBucket()
//count buckets while < index
read part of file in bytes
readPartInputFile()
byte[] fileContByte
double[] bufferDoubles
check if double is in between range of the bucket
checkValidRangeBucket()
//if num >= rangeMin
&& num <= rangeMax
valid number found and belongs to the bucket, add
addExactNumToArr()
double[] bucketExactNums
long[] positionExactNums
sort exact nums retrieved from file
sortExactNums()
double[] bucketExactNums
long[] positionExactNums
find first occurance of the double
findFirstOcc()
find last occurance of the double
findLastOcc()
print number + first and last occ to screen
screen / cmd