README

[Bitwise.java]

set(byte b): use | and relative position's bitmask to set the bit at certain position to 1.
set(byte[]): count from right to left, rolling the certain position by %.
isset(byte b): use & and relative position's bitmask to check if the result >0, if yes, the bit is set.
isset(byte[]): find the right bit first and then use isset(byte b) to check.
clear(byte b): use ~ to inverse the relative position's bitmask, then use & to set bit to 0.
clear(byte[]): find the right bit's position first and then use clear(byte b) to clear.

[MyFileSystem.java]

When the blockNum is > 9, we check if it's > max file size block, if yes, report error, otherwise implement indirect block. Use different level indirect block according to the known blockNum.

If a single indirect block is used, assign a block on disk to it, if no free block, return null. Then update the inode's ptr array, then return a new direct data block according to this indirect block.

The same idea for double and triple, the differences are that they need more pointer positions at each level, and each level may cause "no free space error", and it also needs to update the ptrs in each indirect block.

[Misbehavings]

Hole detection in double and triple indirect blocks are not implemented due to time reason.