Q1(a) We have two direct pointers and one indirect pointer.

Each direct pointer points to a data block, so it has 4KB, totally, we have 8KB.

Each indirect pointer points to a data block with all of its data storing direct pointer, i.e. $\frac{4096 \text{ Byte}}{4 \text{ Byte for each direct pointer}} = 1024 = 1K$

$1K \times 4KB = 4MB$

Finally, we conclude the biggest size of a file with SFS is $4MB + 8KB$.

Q1(bi) read_t(inum, 7000, buf1, 10000);

First, we calculate a and b:

$$a = [7000/4096] = 1$$

$$b = 7000 \bmod 4096 = 2904$$

Because $a<2$, read_t() starts at direct_blk[1] = 9

Then we calculate a' and b':

$$a' = [(2904 + 10000 - 1)/4096] = 3$$

$$b' = (2904 + 10000 - 1) \bmod 4096 = 615$$

Because $a' \geq 2$, read_t() ends at indirect block. The block number is Cell[(3-2)] = Cell[1] = 19

Thus, read_t() will read data blocks 9, 17, 19

Q1(bii) read_t(inum, 12000, buf1, 10000);

$a = [12000/4096] = 2$

$b = 12000 \bmod 4096 = 3808$

read_t() starts at indirect block. The block number is Cell[(2-2)]

$= \text{Cell}[0] = 17$

$a' = [(3808 + 26000 - 1)/4096] = 7$

$b' = (3808 + 26000 - 1) \bmod 4096 = 1135$

read_t ends at indirect block. The block number is Cell[(7-2)]

$= \text{Cell}[5] = 27$

Thus, read_t() will read data block 17, 19, 20, 21, 25, 27

Q1(biii) read_t(inum, 10000, buf1, 36000);

$a = [10000/4096] = 2$

$b = 10000 \bmod 4096 = 1808$

read_t() starts at indirect block. The block number is Cell[(2-2)]

$= \text{Cell}[0] = 17$

$a' = [(1808 + 36000 - 1)/4096] = 9$

$b' = (1808 + 26000 - 1) \bmod 4096 = 943$

read_t ends at indirect block. The block number is Cell[(9-2)]

$= \text{Cell}[7] = 29$

Thus read_t() will read data block 17, 19, 20, 21, 25, 27, 28, 29

Q1(biv) read_t(inum, 1000, buf1, 31000);

$a = \lfloor 1000 / 4096 \rfloor = 0$

$b = 1000 \bmod 4096 = 1000$

read_t() starts at direct_blk[0] = 3

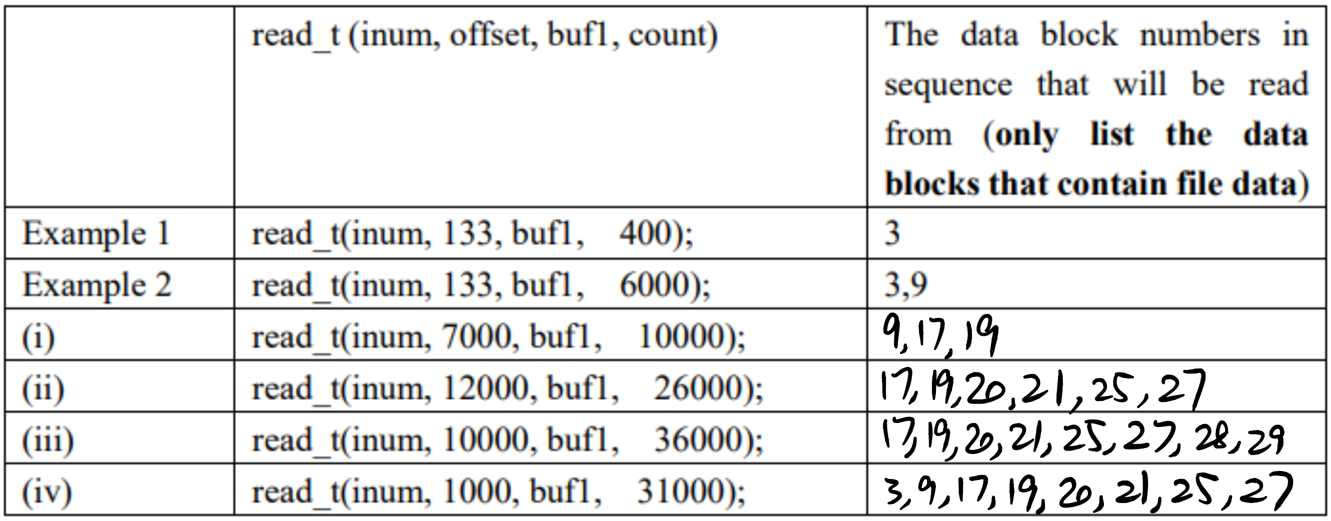
$a' = \lfloor (1000 + 31000 - 1) / 4096 \rfloor = 7$

$b' = (1000 + 31000 - 1) \bmod 4096 = 3327$

read_t ends at indirect block. The block number is cell[(7-2)]

= cell[5] = 27

Thus read_t() will read data block 3, 9, 17, 19, 20, 21, 25, 27



	read_t (inum, offset, buf1, count)	The data block numbers in sequence that will be read from (**only list the data blocks that contain file data**)
Example 1	read_t(inum, 133, buf1, 400);	3
Example 2	read_t(inum, 133, buf1, 6000);	3,9
(i)	read_t(inum, 7000, buf1, 10000);	9, 17, 19
(ii)	read_t(inum, 12000, buf1, 26000);	17, 19, 20, 21, 25, 27
(iii)	read_t(inum, 10000, buf1, 36000);	17, 19, 20, 21, 25, 27, 28, 29
(iv)	read_t(inum, 1000, buf1, 31000);	3, 9, 17, 19, 20, 21, 25, 27

Q2

Data block 0: "/"

.	0
..	0
dir7	7
dir6	6
dir11	11
file5	18

Data block 7: "dir7"

.	7
..	0
dir3	3
dir1	1

Data block 8: "dir8"

.	8
..	11
dir10	10
dir 5	5

Data block 4: "dir4"

.	4
..	3
dir	9

Data block 10: "dir10"

.	0
..	8
file2	15
file4	17

Q2(b)

inode 0 → data block 0 → inode 11 → data block 11

→ inode 8 → data block 8 → inode 10 → data block 10

→ inode 15