Problem 1

1. 4096/190 = 21 records
2. 1000000/21 = 47619.04

Ceil(47619.04) = 47620 blocks

One track has 50/(4096/1024) = 12 blocks,

One cylinder has 12\*4\*2=96 blocks.

We need 47620 blocks to store this file.

So we need 47620/96 = 497 cylinders,

1. Each block can store 4096/100 = 40 records,

Each track can store 12 blocks

1000\*4\*2\*12\*40 = 3840000 records

1. We need 100000/40 = 2500 blocks

2500 / 12 = 209 tracks

209/4/2 = 27 cylinders

Access time = seek time + Rotational Delay + Transfer Time

Sequential Access → We don't need rotational delay

A track can be read/written in one rotation → Transfer Time = no of tracks \* time of one rotation = 209 \* (1/7200\*60) = 1.74 s

Seek time is the time to locate the tracks → Seek time of the file = no Of Cylinders \* seek time of track = 27 \* (1/7200\*60) = 0.225s

Therefore, total access time is 1.74+ 0.225 = 1.97 seconds.

1. For any block,

access time = seek time + rotational delay + transfer time

Seek time = 7 msec

Rotational delay = (1/7200\*60) /2 = 4 msec

Transfer time = 4k/((1\*50/(1/7200\*60))K/sec) = 0.67 msec

The access time for a block of data is 11.67 msec

The file contains 2500 blocks, so the access time is

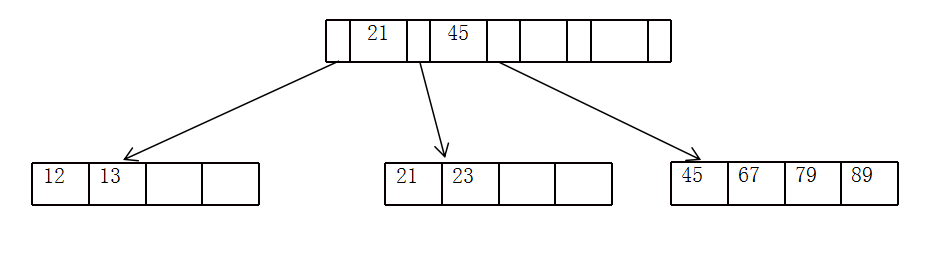
29.175 sec.

Problem 2

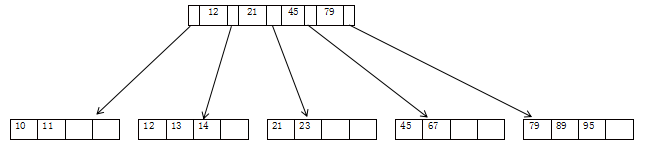
1. step 1

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | 13 | 21 | 45 |

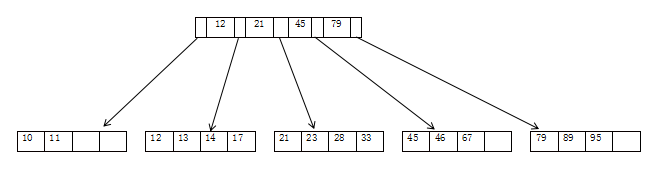
Step 2



Step 3.



Step 4



1. utilization = 20/24 = 0.83
2. yes

10 11 12 13 14 17 21 23 28 33 45 46 47 67 79 89 95

Problem 3

2369, 2428, 4750,

0b1000001

0b1111100

0b1110

6975, 9208, 1821,

0b111111

0b1111000

0b11101

4692, 3943, 1620,

0b1010100

0b1100111

0b1010100

7115, 4871, 5659,

0b1001011

0b111

0b11011

1074, 4981, 1453

0b110010

0b1110101

0b101101

h(K) = K mod 128

65 124 14 63 120 29 84 103 84 75 7 27 50 117 45

0b1000001

0b1111100

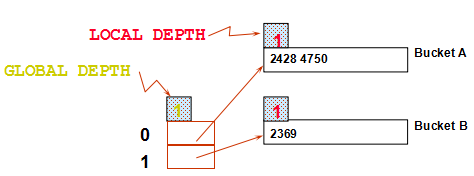
0b1110

0b111111

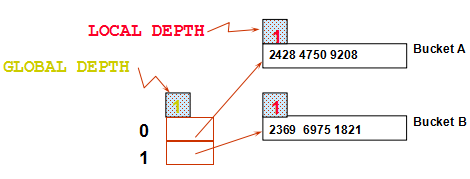
0b1111000

0b11101

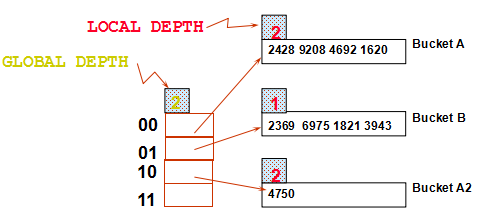
Step 1



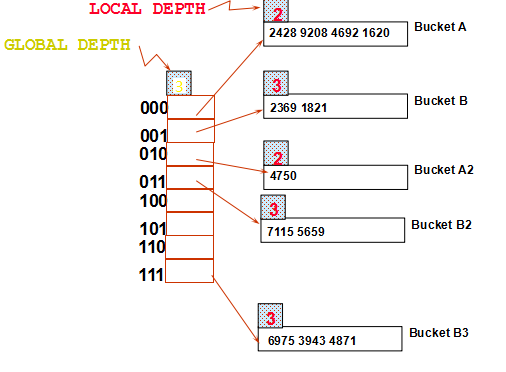
Step 2



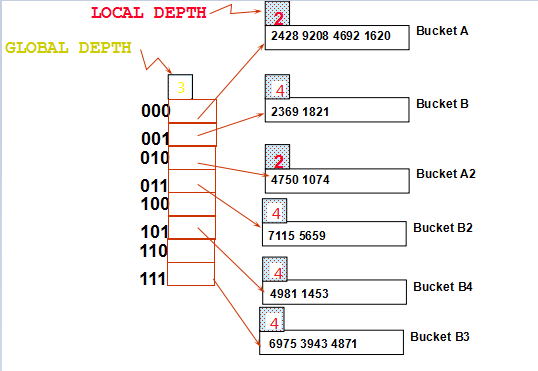
Step 3



Step 4



Step 5



Problem 4

Problem 5

1.

S is smaller, so to be out relation.

Cost of scanning S is 5000/10 = 500 I/Os; a total of 500/5 = 100 blocks.

Per block of S, we scan Reserves; 100\*(20000/10) = 200000 I/Os.

Total cost: 100\*(20000/10) + 500 = 200500 I/Os.

2.

R as inner:

Cost: N + ( (M\*pS) \* cost of finding matching R tuples)

Scan Sailors: 500 page I/Os, 10\*500 tuples.

For each Sailors tuple: 1.2 I/Os to find index page with data entries, plus

cost of retrieving matching Reserves tuples.

Cost of retrieving them is 3 I/Os.

Totals: 500 + 10\*500\*4.2 = 21.5K I/Os

3.

Cost: Sort S +Sort R + (M+N)

4M+4N+M+N = 5M+5N = 5(2000+500) = 12.5K I/Os

4.