RAID

1.

- 10 disks
- 100 GB
- 8 have transfer rate: 1000 Mbits/sec
- 2 have transfer rate: 500 Mbits/sec
- The DB has a stable size of 750GB

RAID 0

Storage effiency: 100%

As we only require 750GB and can access up to 1000GB it gives a redundant 250GB

The average read access time is:

1000 * 8 = 8000Mbits/sec

750gb = 6,144,000mbit

6144000 / 8000 = 768 seconds

RAID 1

Storage effiency: 50% or lower

1000GB * .50 = 500GB (50% avilable harddisk)

750-500=-250 (250GB more is required for a stable DB)

It is not possible to establish a stable DB with the space provided

As only half of the

The average read access time is:

can't implement as the database is too big for the storage space

---

RAID 3

Storage effiency: Medium

1 disk used for control

= 100GB

9 used for storage

= 900GB
= 750GB required for stable DB

900 - 750GB = 150GB excess DB space

so storage effiency is 90%

The average read access time is:

900gb = 921600

921600 / 1125 = <u>819 seconds</u>

RAID 5

Storage Effiency: <u>Exact same as RAID 3</u>

The average read access time is:

921600 / 1125 = <u>819 seconds</u>

RAID 10

Storage effiency: 50% (Low)

first RAID 1 then RAID 0

The average read access time is:

910 seconds + 455 seconds = <u>1365 seconds</u>

RAID 0+1

Storage effiency: 50% (Low)

first RAID 0 then RAID 1

The average read access time is:

455 seconds + 910 seconds = <u>1365 seconds</u>

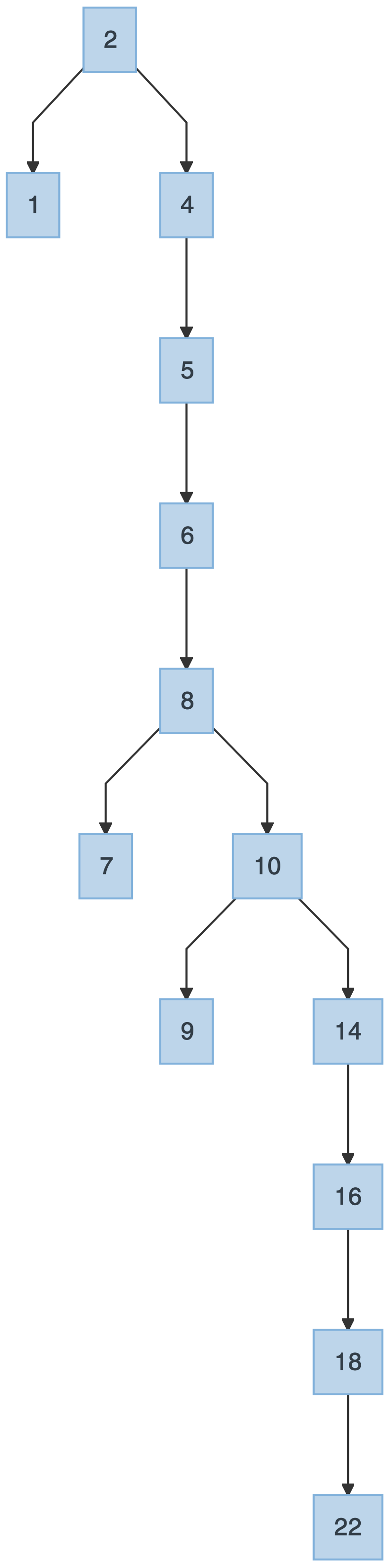
2.

Couldn't figure out how to answer

B-Trees

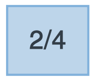
(a) Insert into a simple binary tree (it is not a b-tree, there are no balancing rules) the following data (respecting the order of arrivals):

2,4,5,6,8,10,9,14,16,18,7,22,1

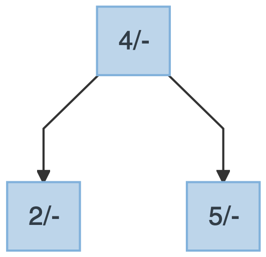


(b) Insert the same data into a B-Tree 2-3 (2 data for each node, 3 pointers, as seen in class).

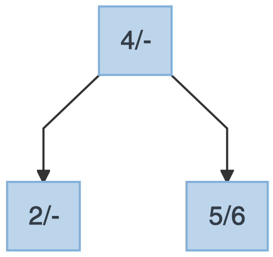
add 2+4



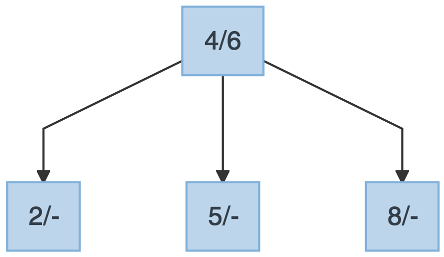
add 5



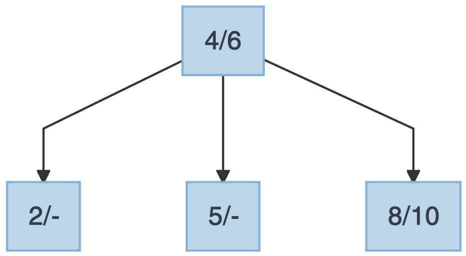
add 6



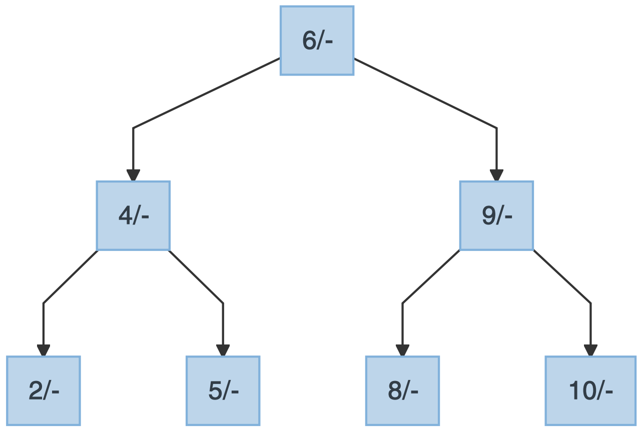
add 8



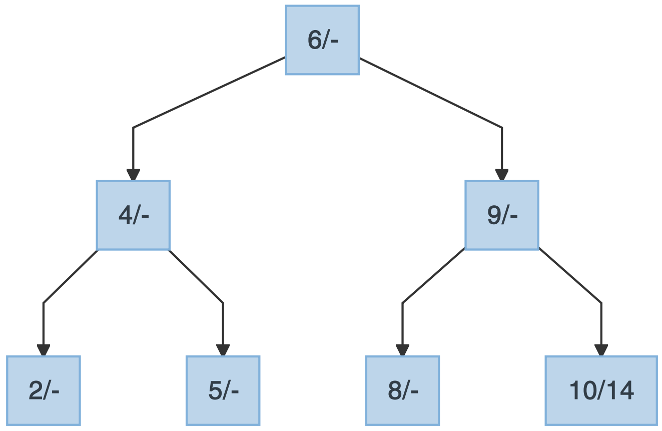
add 10



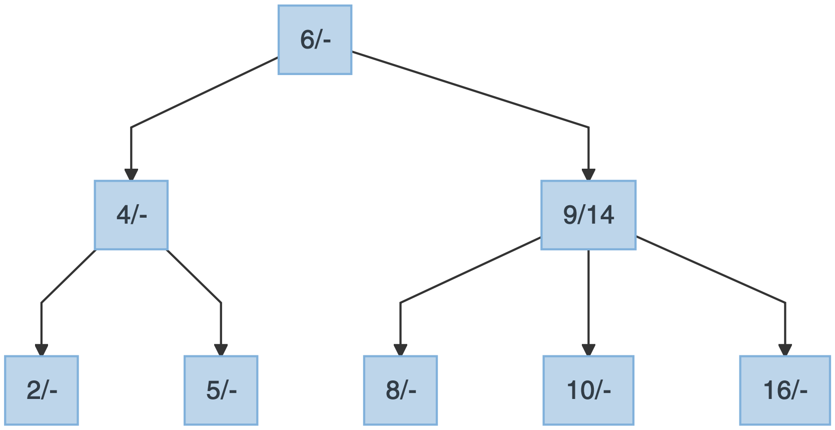
add 9



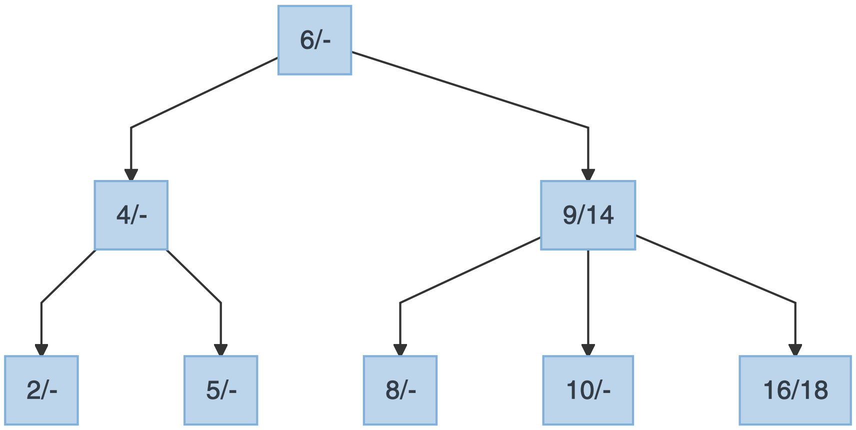
add 14



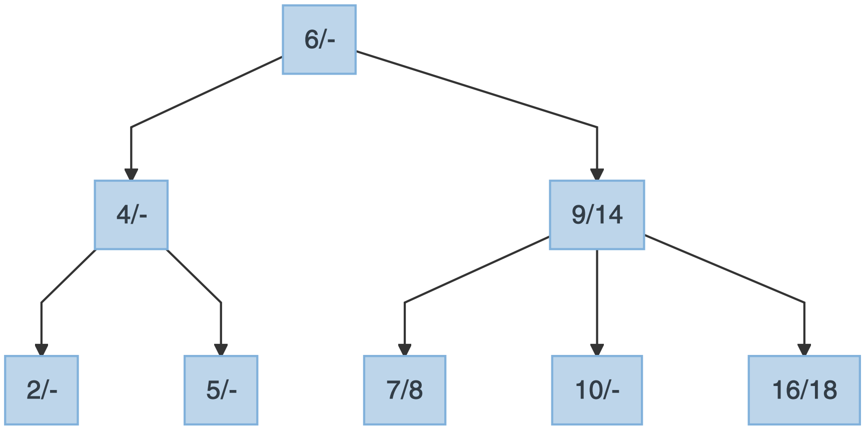
add 16



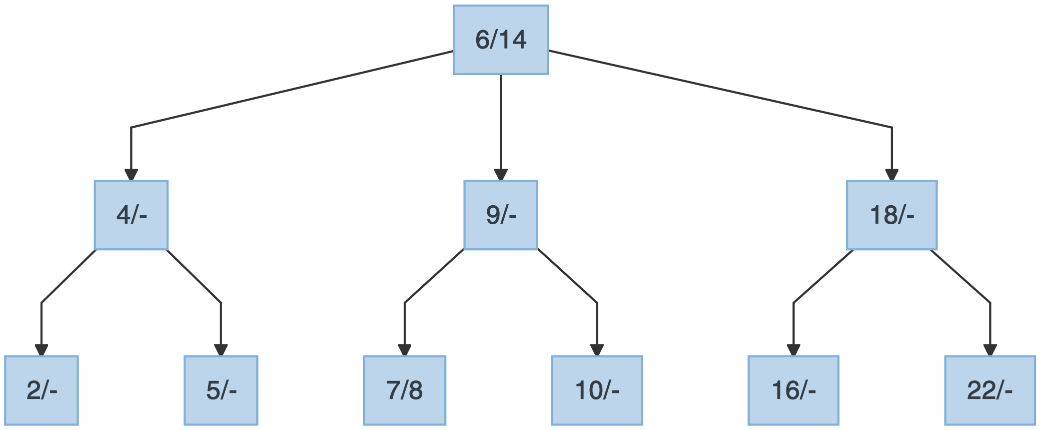
add 18



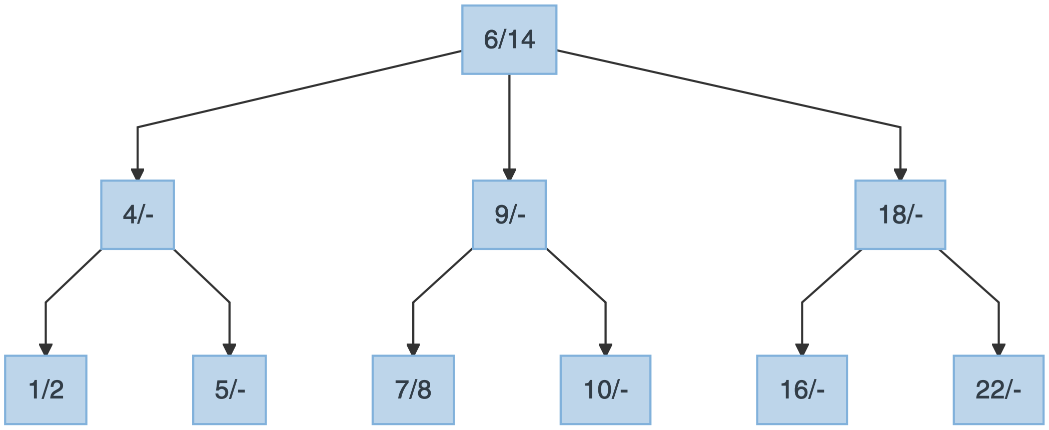
add 7



add 22



add 1



(c) Compare the two trees. Can you see a difference? What is the main advantage of the B-tree?

The Binary Tree in this case is formed into almost an entirely list like structure. This will make make traverse times longer having to move through almost every single node just to get to the bottom most one.

However the B-Tree is arranged into a much more efficient model. Even though it's not entirely free it still makes traverse speed much faster and provides much more structure to the system.