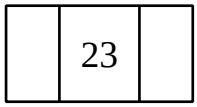
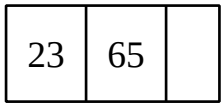
Homework 4

Ben Kero

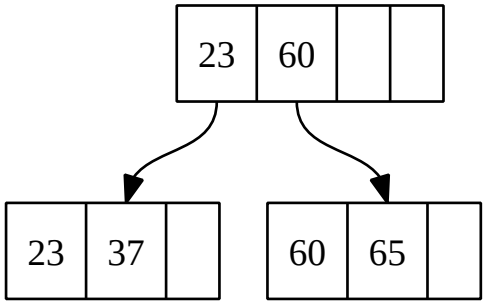
March 5, 2010

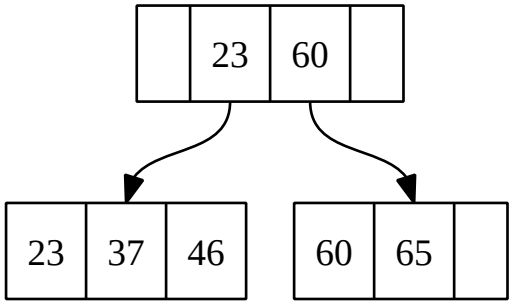
14.15

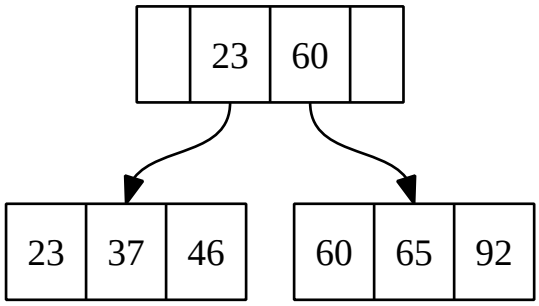
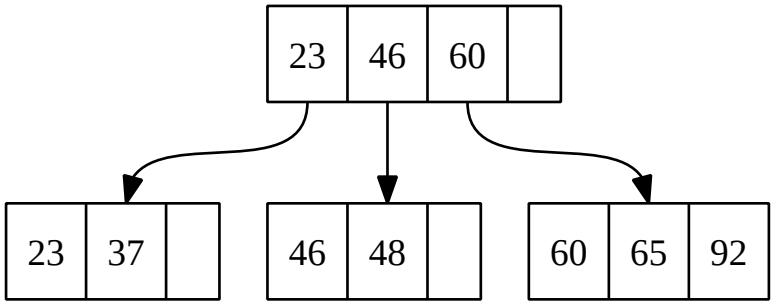
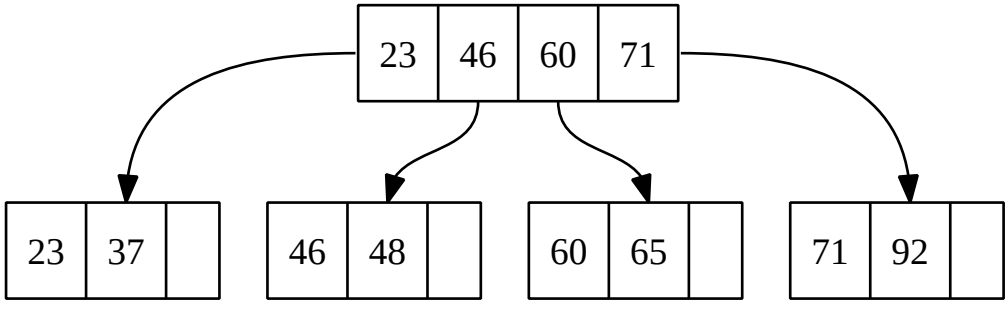
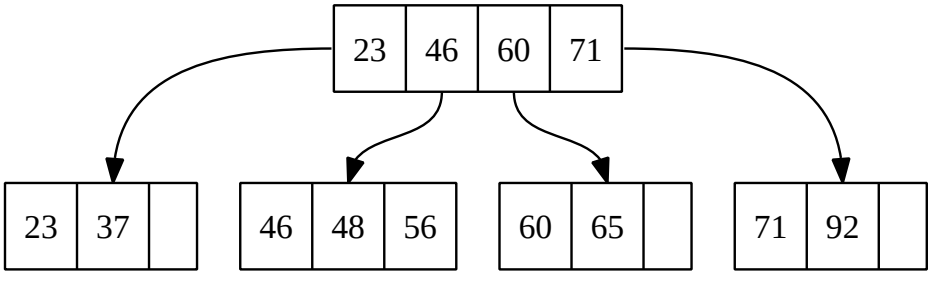
- 

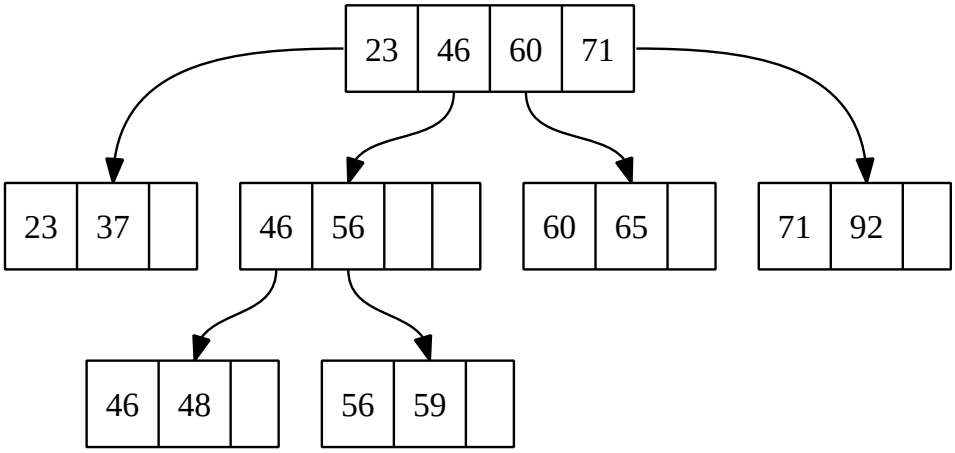
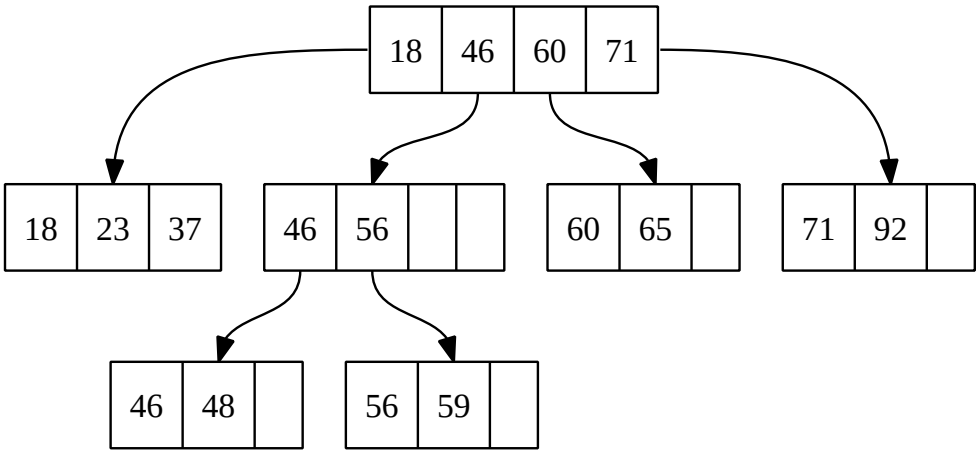
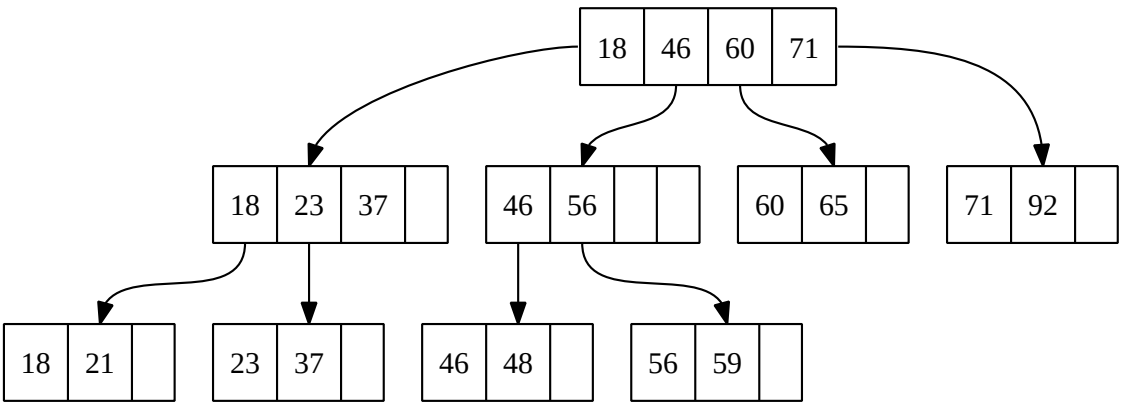
- 

- 

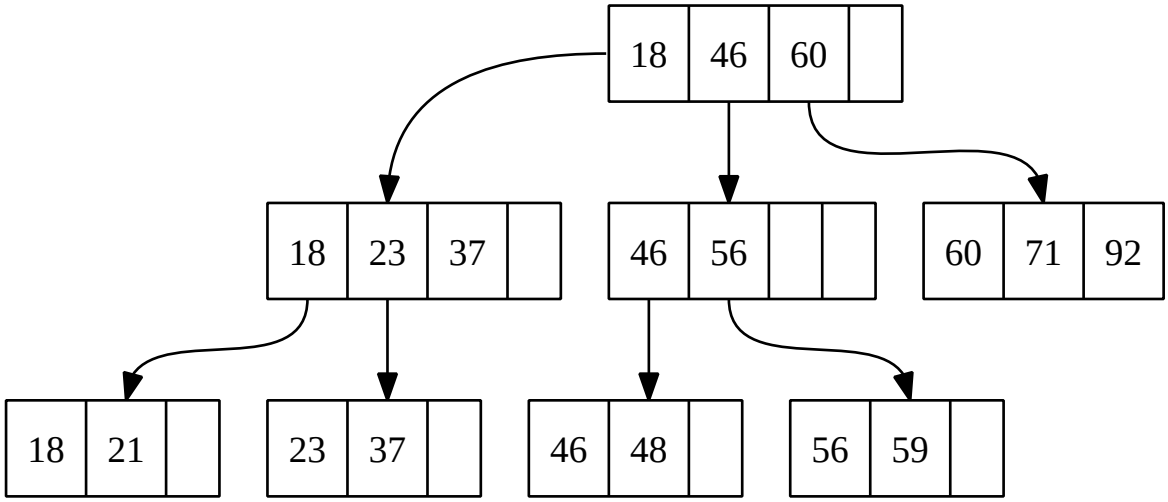
- 

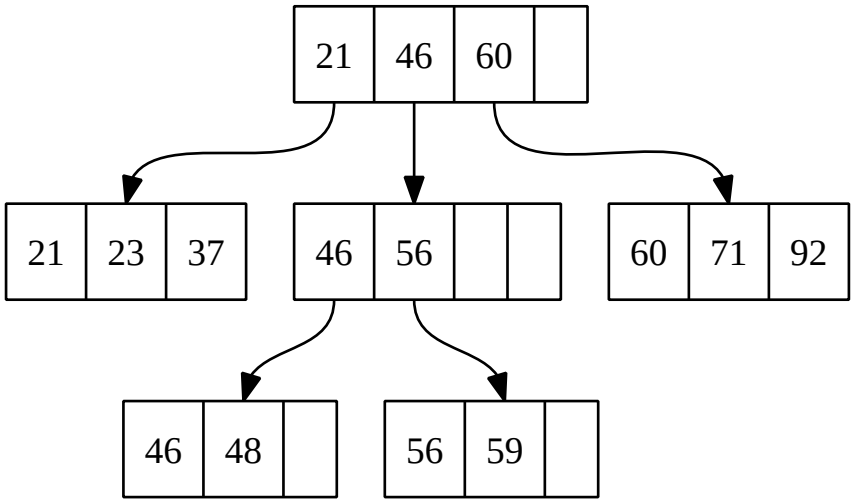
- 

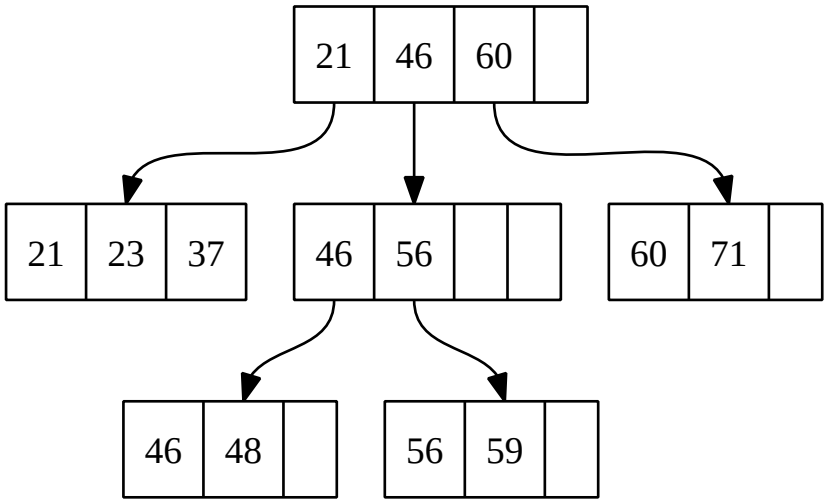
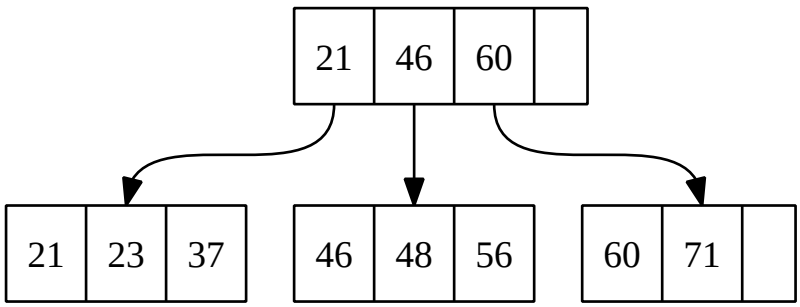
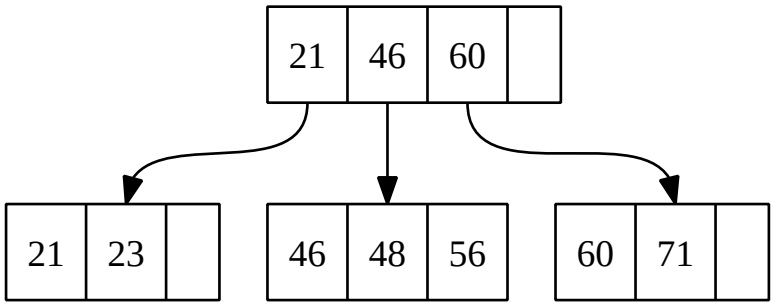
- 
23
60
23
37
46
60
65
92
- 
23
46
60
23
37
46
48
60
65
92
- 
23
46
60
71
23
37
46
48
60
65
71
92
- 
23
46
60
71
23
37
46
48
56
60
65
71
92

- 
23 46 60 71
23 37
46 56
60 65
71 92
46 48
56 59
- 
18 46 60 71
18 23 37
46 56
60 65
71 92
46 48
56 59
- 
18 46 60 71
18 23 37
46 56
60 65
71 92
18 21
23 37
46 48
56 59

14.17

- 
18 46 60
18 23 37
46 56
60 71 92
18 21
23 37
46 48
56 59

- 
21 46 60
21 23 37
46 56
60 71 92
46 48
56 59

- 
21 46 60
21 23 37
46 56
60 71
46 48
56 59
- 
21 46 60
21 23 37
46 48 56
60 71
- 
21 46 60
21 23
46 48 56
60 71

14.19

```
p <- address of top level block of index;
for j <- (step - 1) to 1 do
  begin
    read the index block (at jth index level) whose
      address is p;
    search block p for entry i such that
      Kj(i) exists in K
    search block p for entry i such that
      Kj(i) # K , Kj (i + 1) (if KjCi) is the last
      entry in the block, it is sufficient to satisfy
      Kj(i) # K);
    p <- P_j(i) (* picks the appropriate pointer at
      j^th index level *)
  end;
  read the data file block whose address is p;
  search block p for record with key = K;
```

15.17

No, a nondense index cannot be used in the implementation of an aggregate operator. By definition a nondense index does not contain an entry for each record. It merely contains one for each group of records. Therefore, it is unsuitable for aggregation operators, which require that each individual record count towards some aggregation goal. For example, with a nondense index that contains entries for the the total number of home runs scored in a season of baseball will not help when creating an aggregate operator to find the minimum home runs scored in a game.