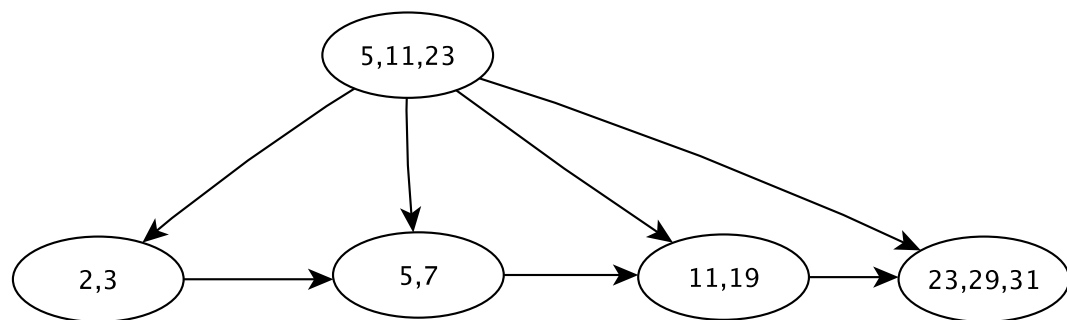


## Assignment 7

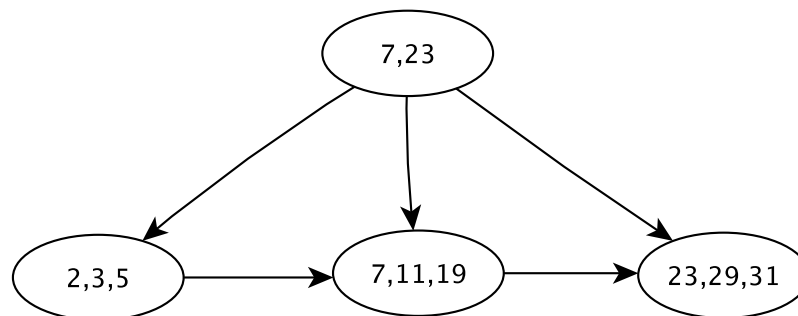
Qinglin Li, 5110309074

## Problem 1

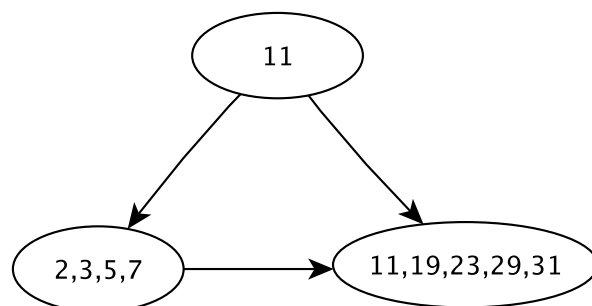
a.



b.



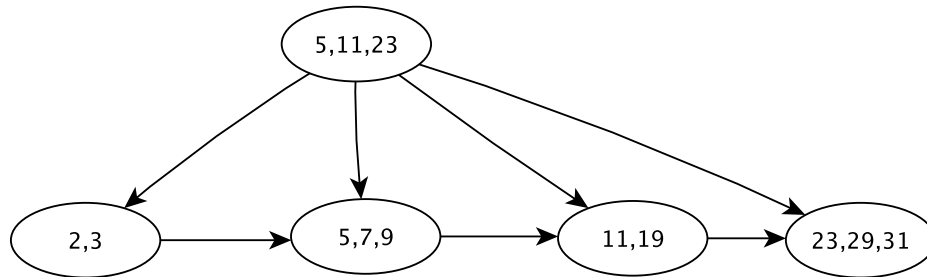
c.



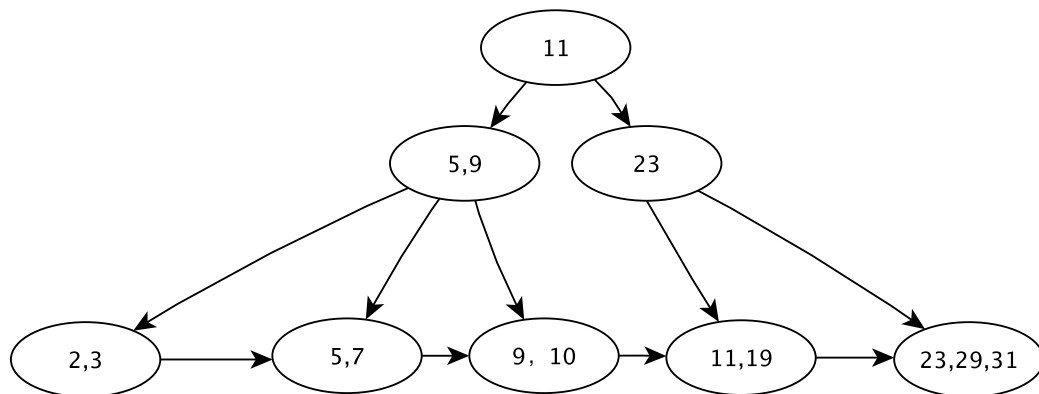
## Problem 2

Four pointers

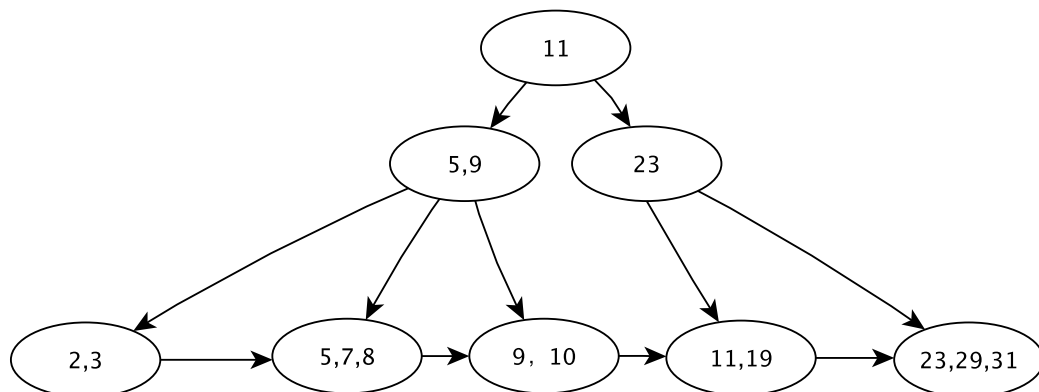
a.



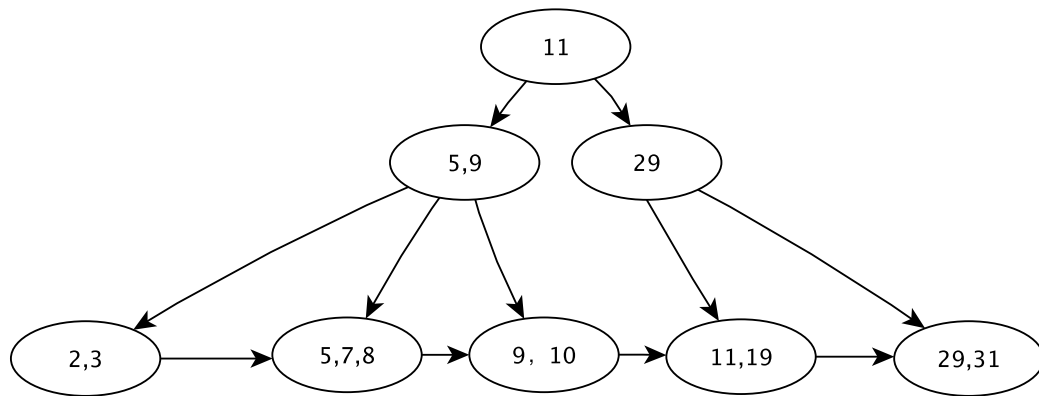
b.



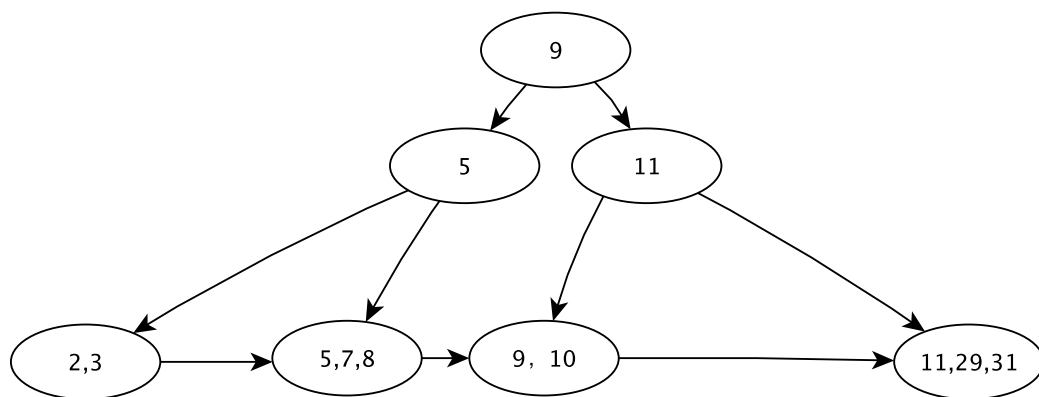
c.



d.

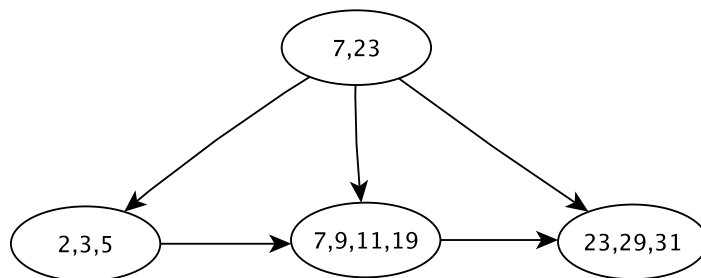


e.

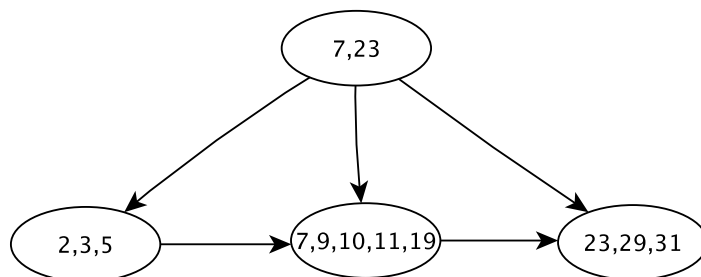


### Six pointers

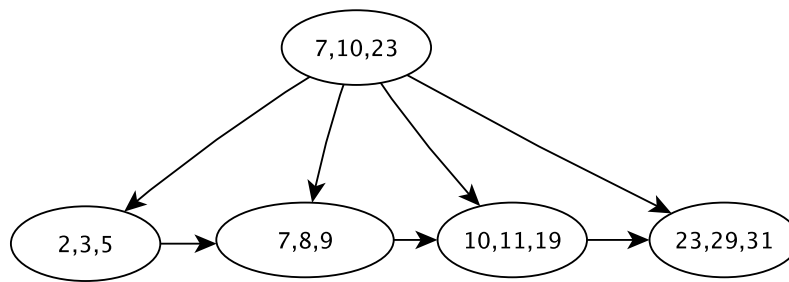
a.



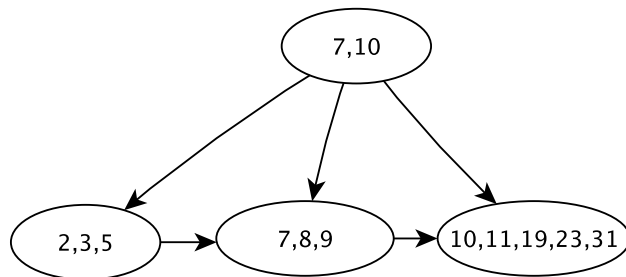
b.



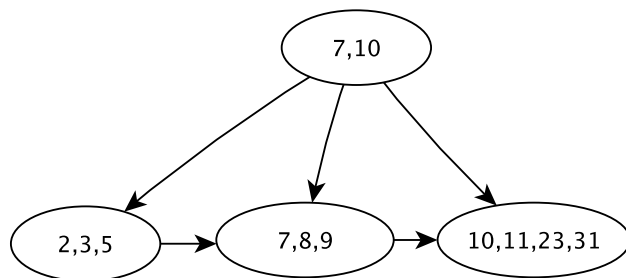
c.



d.

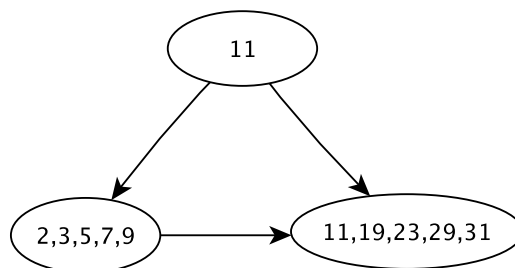


e.

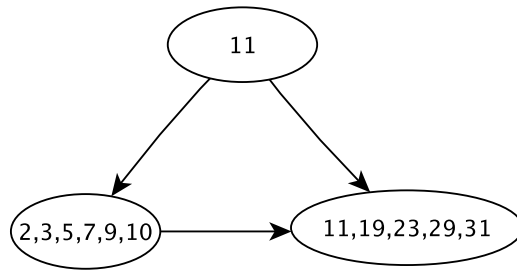


### Eight pointers

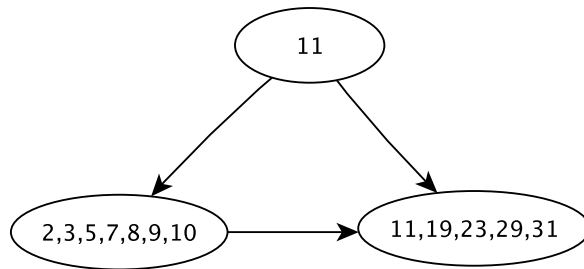
a.



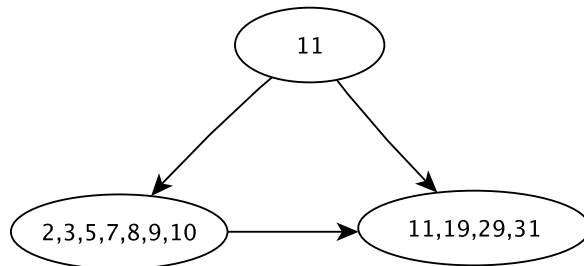
b.



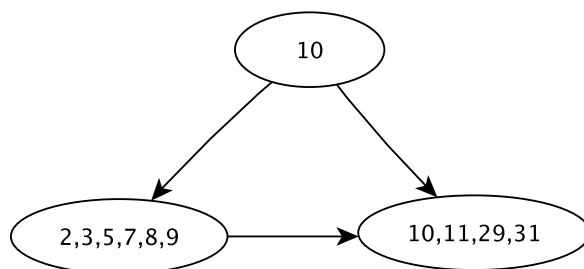
c.



d.



e.



### Problem 3

- Find the first record with  $10 < A < 50$  and enumerate them.  
So  $n_1 + h$  times disk I/O is needed.
- Since we just need to use the condition  $5 < B < 10$  to test each tuple with  $10 < A < 50$ .  
The cost should be the same as **a**
- $n_1 = n_2$

**Problem 4**

- a. Block transfer:  $50000 \times (45000/30) + 50000/25 = 75002000$   
Seek:  $50000 + 2000 = 52000$
- b. Block transfer:  $(45000/30) \times (50000/25) + 50000/25 = 3002000$   
Seek:  $2 \times (50000/25) = 4000$
- c. Since the tuples fit on one block, #Block transfer = #Seek =  $(45000/30) + (50000/25) = 3500$
- d. Assume the memory is large enough.  
#Block transfer is approximately  $(45000/30 + 50000/25) \times 3 = 10500$   
#Seek:  $2 \times (50000/25 + 45000/30) = 7000$

**Problem 5**

- a. Find the first tuple with *building*  $\geq$  “Wastson” and use the index to enumerate them.  
Union these tuples with tuples having *building* = *null*.
- b. Use the techniques as **a** to find tuples with *building* < “Wastson”, *building* > “Wastson”, *building* = *null* and union them all.
- c. Use the techniques as **a** to find all tuples satisfying  $\neg(\textit{building} < \text{“Wastson”})$ .  
Then union them all and test every tuple with condition  $\neg(\textit{budget} < 50000)$ .

**Problem 6**

- a. Using pipelining, output from the sorting operation on *r* is written to a buffer *B*. When *B* is full, the merge-join processes tuples from *B*, joining them with tuples from *s* until *B* is empty. At this point, the sorting operation is resumed and *B* is refilled. This process continues until the merge-join is complete.
- b. If the sortmerge operations are run in parallel and memory is shared equally between the two, each operation will have only  $M/2$  frames for its memory buffer. This may increase the number of runs required to merge the data.