Sample Examination

Last Name (print) ____________________ First Name ____________________

Signature ____________________________ Student Number ________________

There is a total of 100 marks. (180 minutes)

1. a) Specify the following concept.
   - i) conflict operations in a transaction schedule.
   - ii) recoverable schedules.
   - iii) schedule serializability.
   - iv) Conflict equivalence.

   b) Analyze the following transaction execution process. Assume that initially X = 10, Y = 5, N = 5, M = 2. Give the values of X and Y after the concurrent execution of T1 and T2. Show the problem of this computation.

T1	T2
read_item(X);	
X := X – N;	
	read_item(X);
	X := X + M;
write_item(X);	
read_item(Y);	
Y : = Y + N;	
	write_item(X);
write_item(Y);	

Fig. 1

2. Given the following schedules, show which is recoverable, non-recoverable, cascadeless or strict.
   - a) R1(X), W1(X), R2(Z), W2(Z), R1(Y), W2(Y), W2(Y), C1, R2(X), W2(X), C2.
   - b) R1(X), W1(X), R2(X), R1(Y), W2(X), W1(Y), A1, A2.
   - c) R1(X), W1(X), R2(X), R1(Y), W2(X), C2, A1.
   - d) R1(X), R2(X), W1(X), R1(Y), W2(X), C2, W1(Y), C1.
   - e) R1(X), W1(X), R1(Y), W2(Y), W2(X), C1, R2(X), C2.

3. Given the following locking and unlocking operations for two-mode (read-write) locks. Modify the algorithms so that upgrading and downgrading of locks are possible.

```
read_lock(X):
  B:  if Lock(X) = “unlocked”
      then begin Lock(x) ← “read-locked”;
                 no_of_reads(X) ←1
           end
      else ifLock(X) = “read-locked”
           then no_of_reads(X) ← no_of_reads(X) + 1
           else begin   wait(until Lock(X) = “unlocked” and
                        the lock manager wakes up the transaction);
                        go to B
                end

write_lock(X):
  B:  if Lock(X) = “unlocked”
      then Lock(X) ←  “write-locked”
      else begin
                 wait(until Lock(X) = “unlocked” and
                 the lock manager wakes up the transaction);
                 go to B
           end

unlock(X):
  if Lock(X) = “write-locked”
  then begin Lock(X) ← “unlocked”
           wakeup one of the waiting transaction, if any
       end
  else if Lock(X) = “read-locked”
       then begin
                no_of_reads(X) ← no_of_reads(X) – 1;
                if no_of_reads(X) = 0
                then begin Lock(X) = “ unlocked”;
                           wakeup one of the waiting transaction, if any
                     end
            end
```

4. What is the certify lock? Explain why it is needed for multi-version 2PL.

5. How many choices do we have to map an is-a relationship into relation schemas? Please describe these choices. Give the relation schema for the diagram shown in Fig. 2.

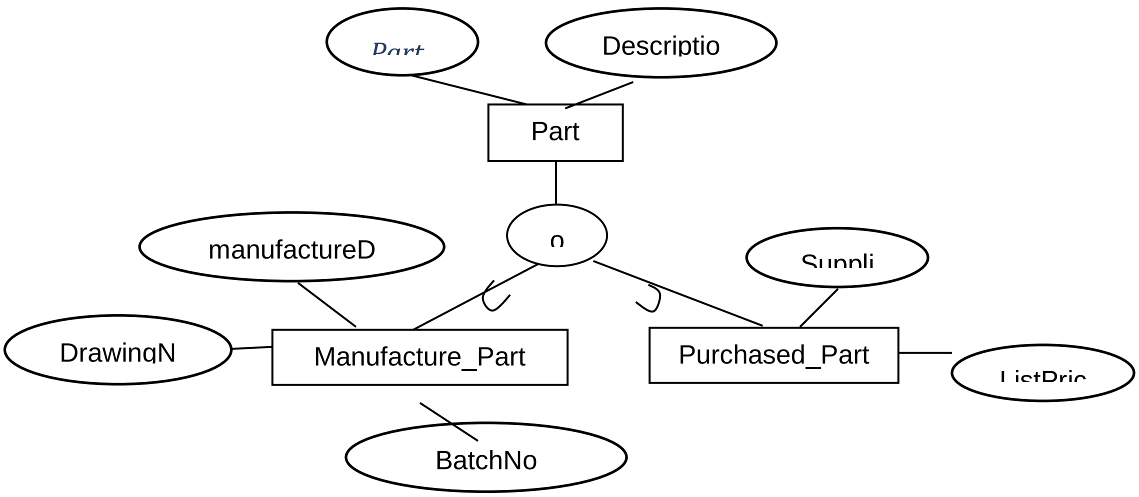


Fig. 2

6. Assume that a database contains only one file which contains two pages and each page contains three recodes, as shown in Fig. 3.

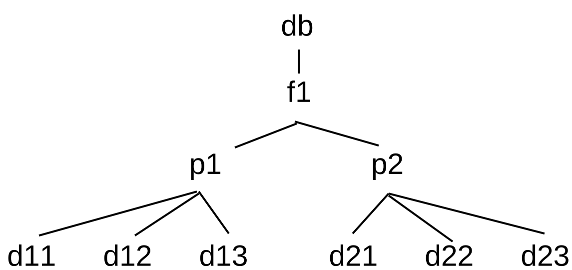
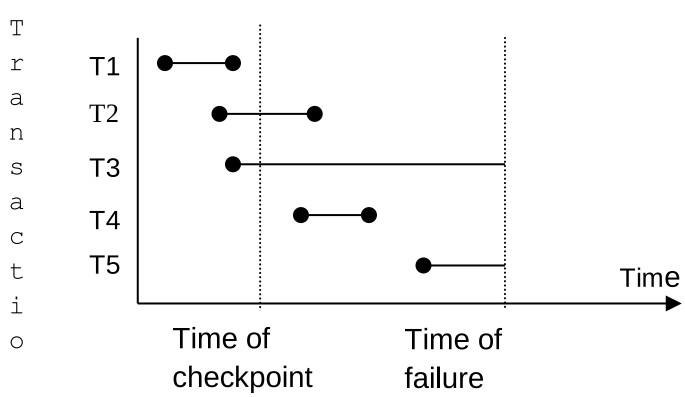


Fig. 3

Suppose transaction T1 wants to update all the records in f1, T2 wants to read d21, d22, and d23, and T3 wants to read only d23. Give a possible serializable schedule for these three transactions in terms of the multiple granularity locking protocol.

7. Consider the five types of transactions given in Fig. 4. If "defered update" stratege is used, which needs to be redone after the crash? Which needs to be undone?



8. a) Describe the heap sorting algorithm.
   b) Apply the quick sorting algorithm to the following sequence and trace the computation process: 1, 10, 9, 8, 5, 7, 6, 2, 4, 3.

9. Discuss the difference between the discretionary access control and the mandatory access control strategies for database security.

10. Given the following spatial database and built-in functions and built-in operators, design an SQL statement to find all the cities which are closer to Limerick than to Dublin.

political-unit:

unit-name	unit-code	unit-population
Republic of Ireland	ie	3.9
Northern Ireland	ni	1.7

boundary:

boundid	bpundpath	unit-code
Republic of Ireland	[(9, 8), (9, 3), (4, 1), (2, 2), (1, 3), (3, 5), (3, 6), (2, 6), (2, 9), (5, 9), (5, 10), (6, 11), (7, 11), (7, 10), (6, 9), (7, 8), (7, 9), (8, 9), (8, 8), (9, 8)]	ie
Northern Ireland	[(7, 11), (9, 11), (10, 9), (10, 8), (8, 8), (8, 9), (7, 9), (7, 8), (6, 9), (7, 10), (7, 11)]	ni

city:

city-name	city-location	unit-code
Dublin	(9, 6)	ie
Cork	(5, 2)	ie
Limerick	(4, 4)	ie
Galway	(4, 6)	ie
Sligo	(9, 6)	ie
Tipperary	(5, 3)	ie
Belfast	(9, 9)	ni
Londondery	(7, 10)	ni

Built-in function: LENGTH(OBJECT). It returns the length of OBJECT.
Built-in operators:

< ->: distance between
<<: is left of ?
<^: is below?
>>: is right of ?
>^: is above of ?

11. Please construct a *kd*-tree over the following data points. Assume that each leaf node can contain two data points.

(25, 60), (45, 60), (50, 75), (50, 100), (50, 120), (70, 110), (85, 140), (30, 260), (25, 400), (45, 350), (50, 275), (60, 350).

12. Construct an R-tree over a set of records for geographical objects with the following coordinates [($x1$, $y1$), ($x2$, $y2$)]:

[(0, 40), (60, 50)] ---- road1
[(40, 0), (60, 40)] ---- road2
[(15, 25), (35, 35)] ---- house1
[(70, 40), (80, 50)] ---- house2
[(70, 5), (80, 15)] ---- house3
[(35, 25), (80, 35)] ---- pipeline

13. Consider the following linear equation:

$$C = \boldsymbol{A}C,$$

where $\boldsymbol{A}$ is a $n \times n$ transition matrix over a web network containing n nodes (each representing a page), and C is a vector containing n variables representing the n pages in G.

Explain why the solution of this equation can used as the estimation of page importance.

Answers

1. a)
   i) a pair of two operations accessing a same database item from two transactions with one of them being a write-item.
   ii) A schedule S is recoverable if no transaction T in S commits until all transactions T' that have written an item that T reads have committed.
   iii) A schedule which is conflict equivalent to a serial schedule.
   iv) Two schedules that have same set of operations and same sat of conflict pairs.

   b) After the concurrent execution of these two transactions, X= 12 and Y = 10. mThe concurrent execution has the 'lost update' problem since change made by T1 is covered by T2. The correct answer should be X = 7 and Y = 10.

2. a) strict
   b) recoverable
   c) non-recoverable
   d) cascadeless
   e) cascadeless

3. Answer:

The followings are the modified algorithms for read_lock, write_lock, and unlock.

Read_lock algorithm

```
read_lock(X, id(T)):
B: if LOCK(X) = "unlocked"
      then begin LOCK(X) ← "read-locked"
            no_of_reads(X) ← 1
            add id(T) to locking_transaction(X)
      end
   else if LOCK(X) = "read-locked"
      then
            no_of_reads(X) ← no_of_reads(X) + 1
            add id(T) to locking_transaction(X)
      end
   else if LOCK(X) = "write-locked"
      then  begin check id(T)
            if (id(T) = locking_transaction(X))
                  then begin LOCK(X) ← "read-locked"
                        no_of_reads(X) ← 1
                  else begin wait(until LOCK(X) = "unlocked" and
                  the lock manager wakes up the transaction);
                  go to B
            end
      end;
```

Write_lock algorithm

```
write_lock(X, id(T)):
B: if LOCK(X) = "unlocked"
      then begin LOCK(X) ← "write-locked"
            add id(T) to locking_transaction(X)
   else begin
      if(no_of_reads = 1 and id(T) = locking_transaction(X))
            then begin LOCK(X) ← "write-locked";
            no_of_reads(X) ← 0
      else begin wait(until LOCK(X) = "unlocked" and the lock
      manager wakes up the transaction);
      go to B
      end;
end;
```

Unlock algorithm

```
unlock(X, id(T))
if LOCK(X) = "write-locked"
      then begin LOCK(X) ← "unlocked"
            remove id(T) from locking_transaction(X)
            wakeup one of the waiting transactions, if any
```

```
      end
else if LOCK(X) = "read-locked"
      then begin
            no_of_reads(X) ← no_of_reads(X) - 1;
            remove id(T) from locking_transaction(X)
            if no_of_reads(X) = 0
                  then begin LOCK(X) = "unlocked";
                        wakeup one of the waiting transactions, if
                  any
                        end
            end;
```

4. Answer:
   In multiversion two-phase locking protocol, certify lock is an exclusive lock issued to confirm that the modification of the currently write-locked item is to be reflected permanently. A certify lock has to be issued on all items that the transaction has hold write lock on, prior to the transaction's commit operation.

The idea behind multiversion two-phase locking protocol is to allow two transactions to hold conflicting locks – read lock and write lock – on the same item, at the same time. The realization of this idea can only be accomplished by allowing two versions for each item X. One version, X, is used to represent the original version of item that has been written by some committed transaction; whereas, the other version, X', is used to represent the copied version of the item that is currently write-locked by a transaction and written by the write operation of the transaction. Therefore, in the multiversion 2PL, one transaction, T, can write the value of X', without any conflict to the value of X that is currently read by some transaction T'.

However, once transaction T is ready to commit, certify lock must come into play. In this protocol, the certify lock act as a confirmation that any changes made on the copied version of item, X', needs to be reflected on the original version of item, X. Hence, once certify locks are acquired on an item, the original version X of the data item is set to the value of version X'. If a certify lock cannot be granted, it shows that some other transactions must be holding a read lock on at least one data item which has been changed by the current transaction. Therefore, the current transaction cannot be committed. It has to wait until all those transactions which are holding the read locks on the relevant data items have been committed.

5. We have altogether 4 choices to handle the is-a (specialization/generalization) relationship.
   a. Generate a table for each class involved in the relationship. They have the same data type for the primary key.
   b. Generate a table for each subclass. But in each table, all the attributes of the superclass are inherited.
   c. This choice is specific for the disjoint subclasses. Generate only table for all the classes with all the attributes being included. In addition, a discriminator attribute is involved.

d. This choice is specific for the overlapping subclasses. Generate only table for all the classes with all the attributes being included. In addition, for each subclass a flag attribute is created.

Part (PartNo, Descr, *Mflag*, DrawingNo, ManDate, BatchNo, *Pflag*, SupName, ListPrice)

6. The following is a possible interleaved execution.

T1:	T1:	T1:
IX(db)		
X(f1)		
write(f1)	IS(db)	
unlock(f1)		
unlock(db)		
	IS(f1)	
	IS(p2)	
	S(d21)	
		IS(db)
		S(f1)
		IS(p2)
		S(d23)
		Read(d23)
		unlock(d23)
		unlock(p2)
		unlock(f1)
	S(d22)	
	S(d23)	
	read(d21)	
	read(d22)	
	read(d23)	
	unlock(d21)	
	unlock(d22)	
	unlock(d23)	
	unlock(p1)	
	unlock(f1)	

7. T1: no redo, no undo
   T2: redo
   T3: no redo, no undo
   T4: undo
   T5: ignore

8.

9. Discretionary access control (DAC)

privileges such as read, write, update are granted to users

a certain amount of discretion is given to the owner of data or anyone else with appropriate authority

Mandatory access control (MAC)

multilevel security is applied to data and users

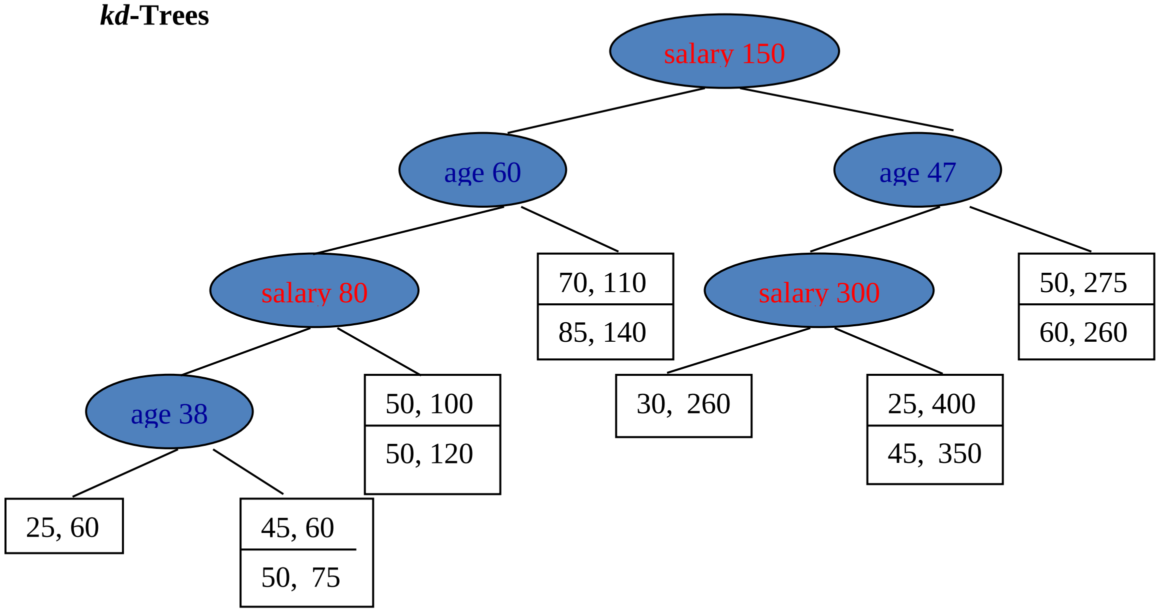
controlled by a central authority, not by owners of an object

the owner/creator of an object does not decide who has clearance to see the object

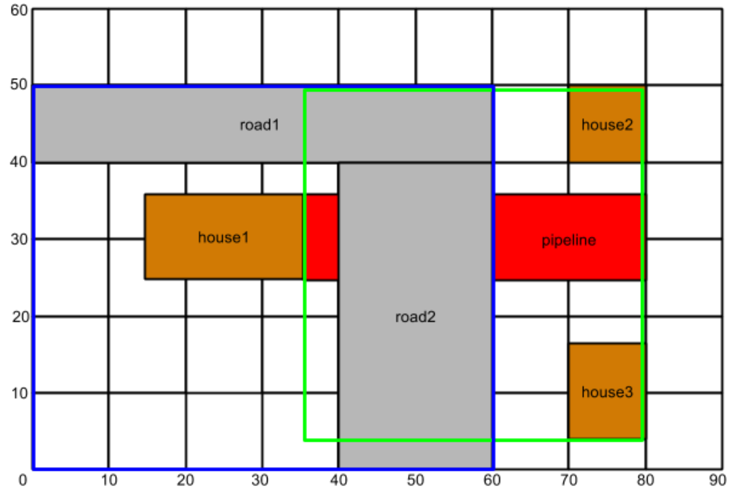
10. SELECT dest.cityname FROM city orig, city dest, city other
WHERE orig.cityname = ‘Limerick’
AND other.cityname=’Dublin’
AND dest.cityname <> ‘Limerick’
AND dest.cityname <> ‘Dublin’
AND (orig.cityloc <-> dest.cityloc) < (dest.cityloc<->other.cityloc)

11.

***kd*-Trees**

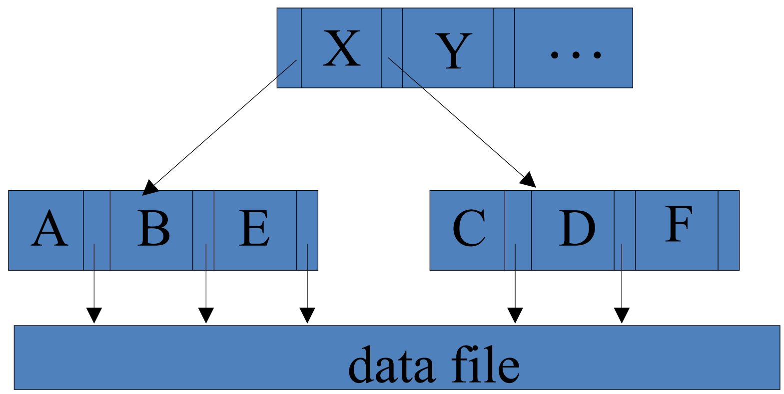


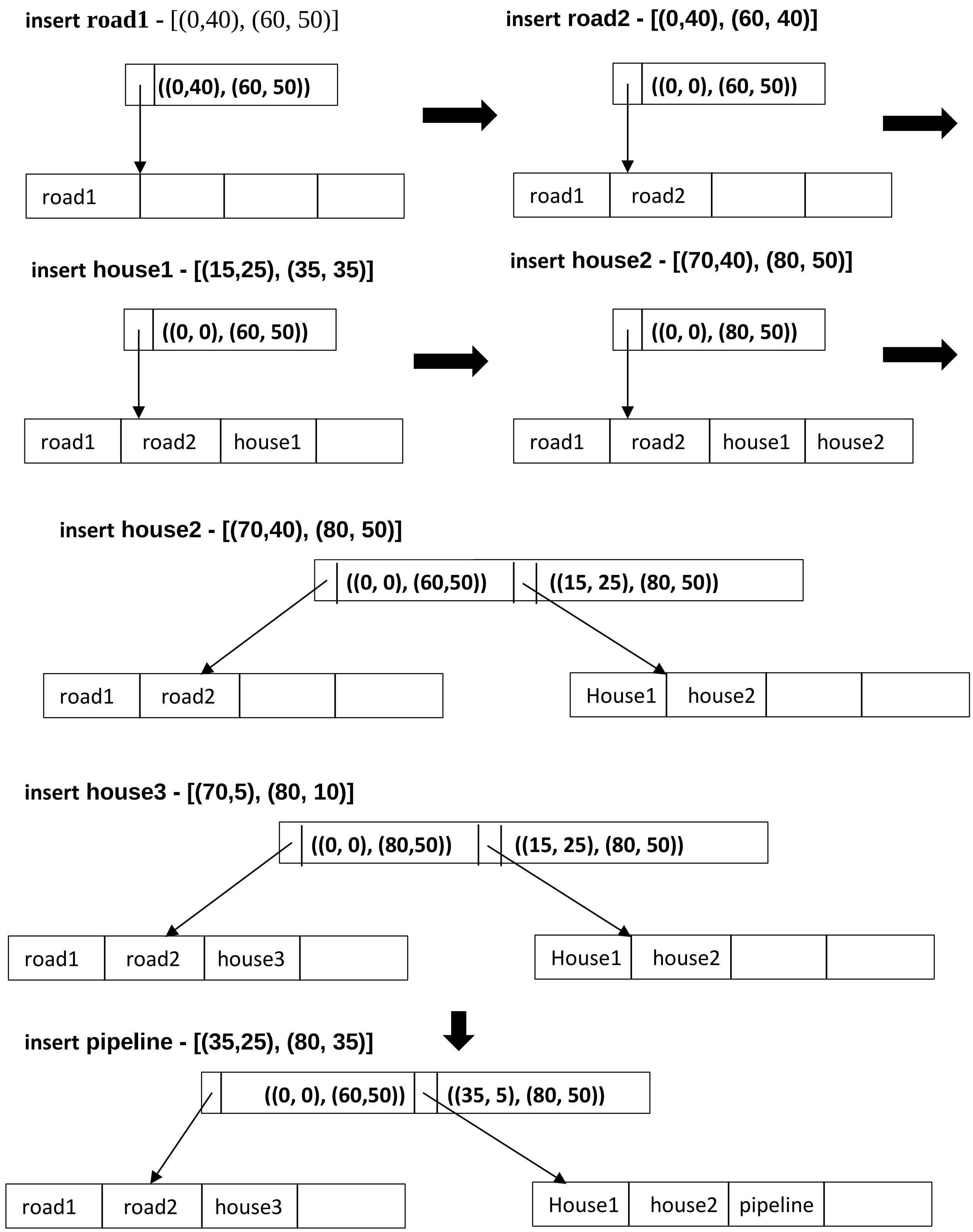
12.



I set the following variables from the picture above:

X = blue rectangle, includes road1, road2 and house1
Y = green rectangle, includes pipeline, house2 and house3
A = road1, B = road2, C = house1, D = house2, E = house3, F = pipeline


X
Y
...
A
B
E
C
D
F
data file


insert road1 - [(0,40), (60, 50)]
((0,40), (60, 50))
road1
insert road2 - [(0,40), (60, 40)]
((0, 0), (60, 50))
road1
road2
insert house1 - [(15,25), (35, 35)]
((0, 0), (60, 50))
road1
road2
house1
insert house2 - [(70,40), (80, 50)]
((0, 0), (80, 50))
road1
road2
house1
house2
insert house2 - [(70,40), (80, 50)]
((0, 0), (60,50))
((15, 25), (80, 50))
road1
road2
House1
house2
insert house3 - [(70,5), (80, 10)]
((0, 0), (80,50))
((15, 25), (80, 50))
road1
road2
house3
House1
house2
insert pipeline - [(35,25), (80, 35)]
((0, 0), (60,50))
((35, 5), (80, 50))
road1
road2
house3
House1
house2
pipeline

13. By solving the equation, we get in fact the probabilities to access a page in the web network. They can be used as the estimation of page importance sine more important, the higher the access probability.