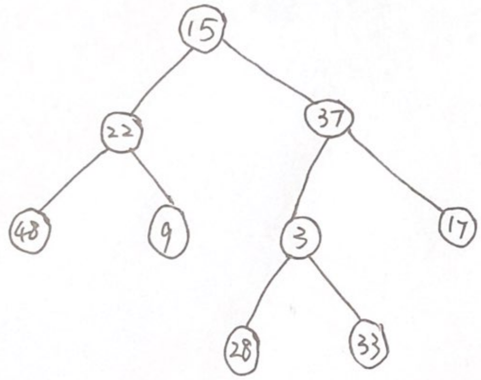
1155154043 LU, Hui

Q1. I.



II. We can use binary ~~seach~~ search ~~to~~ examine the midpoint recursively to see if it's larger or smaller than or equal to the element on the same position in array [1, 2, ..., n]. If the element is larger, meaning the missing element is in the left, otherwise it should be in the right half of the array. The pseudo-code is as follows.

```
int findmissing ( int A[], int n){
        int b = 0, e = n-1;
        while (s < e){
            int mid = (b+e)/2;
            if ( A[mid] <= mid+1){
                s = mid+1; }
            else {
                e = mid-1; }
        }
        if (A[S] > s+1){return S+1;}
        else {: return A[S]+1; }
}
```

III. Apply the dynamic programming method. ~~For each node.~~

Q2. I. Note that 1 page can hold $\frac{2K}{4} = \frac{2^{11}}{2^2} = 2^9$ page entries.

The 32-bit address space has $\frac{2^{32}}{2^{11}} = 2^{21}$ pages in total.

Thus we need 3 levels of page tables, with one level of page table containing 2^3 entries, while the other 2 levels of pages tables containing 2^9 page entries each.

II. 1, 2, 1, 2, 3, 2, 2, 4, 5, 4, 2, 5, 4, 2, 5, 4.

III. (1). abc should be set to 1.

(2) xyz can be set to 0.

(3). The disadvantages ~~is that~~ are that the spin-wait is a waste of CPU time, and some process may wait forever, which may cause starvation.

To avoid the stravation we can set a queue to hold the processes and run them in order.

I. a $\quad P \leftarrow \pi_{PID,\,Category}(\sigma_{category = \text{"Toy"}}(PRODUCT))$

$C \leftarrow \pi_{CID,\,Address}(\sigma_{address = \text{"shatin"}}(CUSTOMER))$

$Result \leftarrow \pi_{EID,\,Name}(TRANSACTION \bowtie EMPLOYEE \bowtie P \bowtie C).$

b. $\cancel{((_{EID}\mathcal{G}_{sum(price * Quantity)}(TRANSACTION)) \bowtie CUSTOMER)}$

$R \leftarrow {}_{EID}\mathcal{G}_{sum(price * Quantity)\ as\ money}(TRANSACTION)$

$Result \leftarrow \pi_{EID,\,NAME,\,money}(R \bowtie EMPLOYEE).$

II. a.

```
select EID, EMPLOYEE.Name
from   TRANSACTION, EMPLOYEE, CUSTOMER, PRODUCT
where  TRANSACTION.EID = EMPLOYEE.EID
   and TRANSACTION.PID = PRODUCT.PID
   and TRANSACTION.CID = CUSTOMER.CID
   and CUSTOMER.Address = "Shatin"
   and PRODUCT.Category = "Toy".
```

b.

```
Select T.EID, E.Name, T.money
from  (select EID, sum(Price * Quantity) as money
       from TRANSACTION
       group by EID) as T, EMPLOYEE as E
where  T.EID = E.EID.
```

Q3. III. a. Note that $N_{employee} = 2000$, $B_{employee} = \frac{2,000}{100} = 20$.

$N_{transaction} = 100,0000$, $B_{transaction} = \frac{100,0000}{50} = 20,000$

For the nested-loop join, if use EMPLOYEE as the outer relation, we need ~~Bemployee~~ $N_{employee} \times B_{transaction} + B_{employee} = 40,000,020$. block accesses; while using TRANSACTION as the outer relation. requires $N_{transaction} \times B_{employee} + B_{transaction} = 20,020,000$ block accesses.

Thus we should use TRANSACTION as the outer relation.

For the block nested-loop join:

EMPLOYEE outer: $B_{employee} + B_{transaction} = 20,020$.

since $B_{transaction} > M = 50$.

Thus EMPLOYEE should be chosen as the outer relation.

Q3 III. (b.) Nested - loop join:

$$N_{transaction} \times B_{employee} + B_{transaction} = 20,020,000$$

~~Nested-lo~~

Block nested - loop join:

$$B_{employee} + B_{transaction} = 20,020.$$

Q4. I. (a). $\{a\}$: sup = 5, $\{b\}$: sup = 4, $\{c\}$: sup = 4, $\{e\}$: sup = 3

$\{a,b\}$: sup = 4. $\{a,c\}$: sup = 4, $\{a,e\}$: sup = 3, $\{b,e\}$: sup = 3

~~$\{a,b,c\}$: sup = 3, $\{a,b,e\}$ sup = 3~~.

$\{b,e\}$: sup = 3,

$\{a,b,c\}$: sup = 3, $\{a,b,e\}$: sup = 3.

(b). $\{a\} \to \{b\}$: $conf(\{a\} \to \{b\}) = \dfrac{sup(\{a,b\})}{sup(\{a\})} = 0.8$

$\{a\} \to \{c\}$: $conf(\{a\} \to \{c\}) = 0.8$

$\{b\} \to \{c\}$: $conf(\{b\} \to \{c\}) = 0.75$.

$\{b\} \to \{e\}$: $conf(\{b\} \to \{e\}) = 0.75$

$\{b\} \to \{a,c\}$: $conf(\{b\} \to \{a,c\}) = 0.75$

$\{b\} \to \{a,e\}$: $conf(\{b\} \to \{a,e\}) = 0.75$.

$\{c\} \to \{a\}$: $conf(\{c\} \to \{a\}) = 1.0$.

$\{c\} \to \{b\}$: $conf(\{c\} \to \{b\}) = 0.75$

$\{c\} \to \{a,b\}$: $conf(\{c\} \to \{a,b\}) = 0.75$

$\{e\} \to \{a\}$: $conf(\{e\} \to \{a\}) = 1.0$

$\{e\} \to \{b\}$: $conf(\{e\} \to \{b\}) = 1.0$.

$\{e\} \to \{a,b\}$: conf = 1.0.

$\{a,b\} \to \{c\}$: conf = 0.75, $\{a,b\} \to \{e\}$: conf = 0.75

$\{a,c\} \to \{b\}$: conf = 0.75, $\{a,e\} \to \{b\}$: conf = 1.0.

$\{b,c\} \to \{a\}$: conf = 1.0, $\{b,e\} \to \{a\}$: conf = 1.0.

Q 4. (II). Model overfitting is ~~[illegible]~~ a case in which the model is ~~[illegible]~~ too complicated than the training data, thus the model can very accurately fit to the training data without the ability of generalization.

Method 1: Improve the size of the trainging data, or reduce the complexity of the model, (e.g. the number of ~~par~~ learnable parameters in the model).

Method 2: Introduce regularization term in the loss function ~~to~~ restrict the model to overfit to the training data.

(III). (a) Iter 1: Allocation: cluster 1: $\{1\}$

cluster 2: $\{3, 4, 6, 8\}$

cluster 3: $\{13, 15, 17, 20, 25\}$.

centroids update: cluster 1: $\mu_1 = 1$

cluster 2: $\mu_2 = \frac{3+4+6+8}{4} = \frac{21}{4}$

cluster 3: $\mu_3 = \frac{13+15+17+20+25}{5} = 18$

Iter 2: Allocation: cluster 1: $\{1, 3\}$.

cluster 2: $\{4, 6, 8\}$

cluster 3: $\{13, 15, 17, 20, 25\}$.

centroids update: $\mu_1 = \frac{1+3}{2} = 2$.

$\mu_2 = \frac{4+6+8}{3} = 6$

$\mu_3 = 18$

Iter 3 : Allocation = cluster 1 : $\{1, 3, 4\}$

cluster 2 : $\{6, 8\}$

cluster 3 : $\{13, 15, 17, 20, 25\}$

centroids : $\mu_1 = \frac{1+3+4}{3} = \frac{8}{3}$

$\mu_2 = \frac{6+8}{2} = 7.$

$\mu_3 = 18$

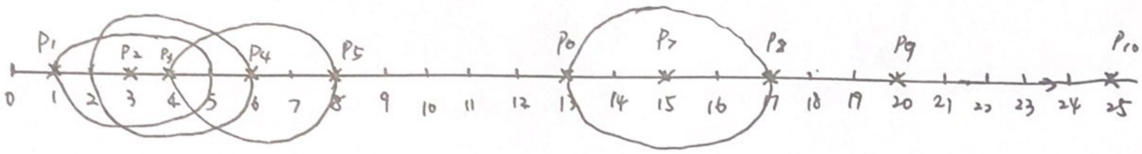
Iter 4 : Allocation : cluster 1 : $\{1, 3, 4\}$

cluster 2 : $\{6, 8\}$

cluster 3 : $\{13, 15, 17, 20, 25\}$

terminates !

III. (b)



core points: P_2, P_3, P_4 P_7.

border points: P_1, P_5, P_6, P_8.

noise points: P_9, P_{10}.

P_1's ε-neighbor: P_2.

P_2's ε-neighbor: P_1, P_2, P_3

P_3's ε-neighbor: P_2. P_3. P_4.

P_4's ε-neighbor: P_3. P_4, P_5.

P_6's ε-neighbor: P_6. P_7,

P_7's ε-neighbor: P_6. P_7. P_8

P_8's ε-neighbor: P_7.

P_9's ε-neighbor: P_9

P_{10}'s ε-neighbor: P_{10}.

final clusters: cluster 1: $\{P_1, P_2, P_3, P_4\ P_5\}$

cluster 2: $\{P_6, P_7, P_8\}$.

Q5. (I)(a) The minimum number of elements in total the method visit is

$$\min(M_1, M_2) + 1.$$

In this case all elements in the longer sequence are larger than those in the shorter one.

sample : List 1 : $1 \to 2 \to 3$

List 2 : $4 \to 5 \to 6 \to 7 \to 8$.

(b). $M_1 + M_2$. In this case the last element in the shorter list is larger than the last element in the longer one.

sample: List 1 : $1 \to 7 \to 11$

List 2 : $2 \to 4 \to 6 \to 8 \to 10$.

(II.). There are two classes of data below, denoted by cross and dot, respectively. The bold cross and dot are their corresponding class prototypes. The dotted line is the Rocchio classification boundary. While the bold star "*" is closer to the centroid of the class 1,

it is more reasonable to classify it as class 2. In this case the 3-NN method is more suitable.

III. $P(X|C)$ should equal to the fraction of times in which word x appears among all words in documents of class c.

$$P(x=w \mid c=c_j) = \frac{N(x_i=w)+1}{N(\text{words in } c_j) + N_w}$$

where $N(x_i=w)$ is the number of times w appears in documents of class c_j, and $N(\text{words in } c_j)$ is the total number of words in documents of class c_j, N_w is the size of the dictionary.