IS2020 Jinchao Li 1155133496

Q1 (DS & Algo.)

I.

```
int isSumTree (struct TreeNode* node){
    int ls, rs;
    if (node == NULL || (node -> left == NULL && node -> right == NULL)){
        return 1;
    }
    ls = sum(node -> left);    // function "sum" is writed below.
    rs = sum (node -> right); // get sum of left for right subtrees.
    if ((node -> value == ls + rs) && isSumTree (node -> left) &&
        isSumTree (node -> right)) { return 1; }
    return 0;
}
int sum (struct TreeNode* node){
    if (node == NULL) { return 0; }
    return sum (node -> left) + node -> value + sum (node -> right);
}
```

(move to top)

II. T:

0	1	2	3	4	5	6	7	8	9	10
11	12	13		33	5		24	27	9	

search for 44 : $44 \% 11 = 0$, examine '0', continue

$[0 + 1 \times (1 + 44 \% 5)] \% 11 = 5$, examine '5', continue.

$[0 + 2 \times (1+4)] \% 11 = 10$, examine '10', break.

∴ locs and keys: 0 → 5 → 10.

III. we can achieve it in a recursive approach:

1: devide A_1, A_2 into two parts respectively. denoted as $A_{1l}, A_{1r}, A_{2l}, A_{2r}$.

2: A_{1l}, A_{2l} are left part, and A_{1r}, A_{2r} are right part.
make sure that len(left) = len(right), max(left) ≤ min(right).
then we have median = $\frac{1}{2}$ [max(left) + min(right)].

3: note that left and right part contains also two sorted subarrays,
So we can repeat 1,2 until get median.

The complexity is $O(\log(n_1+n_2))$. □

IS 2020 Jinchao Li 1155133496

Q2

I. ① Deadlock occurs conditions of : 1. Mutual exclusion, 2. Hold and wait, 3. No preemption, 4. Circular wait.
It occurs when each process holds a resource and wait for another resource held by any other process.
② Starvation occurs when high priority processes keep executing and low priority processes get blocked for ~~inf~~ indefinite time.

II. ① virtrual memory address refers to the virtual store viewed by processes.
② physical memory address refers to hardware addresses of physical memory.
③ there are independent, and virtual one 'map' to physical one.

III. (a) 3 or 8.

(b)

```
bool atomic_add(&word, value){
    compare_and_swap(word, 1, 1);
    word = word + value;
    return word;
}
```

IV. sequence: 0, 2, 3, 7, 1, 3, 2, 3, 7, 6, 5, 3, 2, 6, 5, 6.

LRU:	0	0	0	7	7	7	2	2	2	6	6	6	2	2	2	2
		2	2	2	1	1	1	1	7	7	7	3	3	3	5	5
			3	3	3	3	3	3	3	3	5	5	5	6	6	6
				F	F		F		F	F	F	F	F	F	F	

→ page faults: 10

FIFO	0	0	0	7	7	7	7	3	3	3	5	5	5	6	6	6
		2	2	2	1	1	1	1	7	7	7	3	3	3	5	5
			3	3	3	3	2	2	2	6	6	6	2	2	2	2
				F	F		F	F	F	F	F	F	F	F	F	

→ page faults: 11

∴ page faults of LRU: 10
∵ of FIFO: 11. □

IS 2020 Jinchao Li 1155133496

~~Q7~~ Q3 (Database)

(a)

I. $_{CID}\mathcal{G}_{COUNT\text{-}DISTINCT(MID)>=100}$ (MATCH)

II. $_{PID}\mathcal{G}_{SUM(P1)+SUM(P2)}$ (PLAYER $\bowtie$ MATCH).

(b) I.

select CID
from MATCH
group by ~~distinct~~ MID
having ~~distinct~~ count_distinct (MID) >= 100.

II.

select PID , sum(P1) + sum(P2)
from PLAYER natural join MATCH
where PLAYER.PID = MATCH.P1 or PLAYER.PID = MATCH.P2

(c) COURT: $n_c = 1000$, $nb_c = 50$ $\therefore b_c = \frac{1000}{50} = 20$ (number of blocks)

MATCH: $n_M = 50{,}000$, $nb_M = 40$ $\therefore b_M = 50.000/40 = 1250$

20 entries/node

COURT as outer-relation: $h(B+) = \log_{20/2}(50{,}000) \approx 5$.

$\therefore$ nb of block transfers:

$$b_c + n_c(h(B+)+1) = 20 + 1000 \times (5+1) = 6020.$$

Q4 (Data Mining)

(i) (a) original Gini index $o = 1 - (\frac{5}{8})^2 - (\frac{3}{8})^2 \approx 0.47$

if choose A:

	A=1	A=0
+	3	2
−	0	3

$$\text{Gini\_index}(A) = \frac{3}{8}(1-1) + \frac{5}{8}\left[1 - (\frac{2}{5})^2 - (\frac{3}{5})^2\right]$$

$= 0.3$

gain (A) $= 0.47 - 0.3 = 0.17$

if choose B:

	B=1	B=0
+	2	3
−	2	1

$$\text{Gini\_index}(B) = \frac{4}{8}(1 - 0.5^2 - 0.5^2) + \frac{4}{8}\left[1 - (\frac{3}{4})^2 - (\frac{1}{4})^2\right]$$

≈ 0.44

gain(B) $= 0.47 - 0.44 \approx 0.03 <$ gain (A).

$\Rightarrow$ choose A as first splitting criteria.

(b) $P(+) = \frac{5}{8}$, $P(-) = \frac{1}{8}$, $P(A=1|+) = \frac{3}{5}$, $P(A=0|+) = \frac{2}{5}$, $P(A=1|-) = 0$, $P(A=0|-) = 1$.

$P(B=1|+) = \frac{2}{5}$, $P(B=0|+) = \frac{3}{5}$, $P(B=1|-) = \frac{2}{3}$, $P(B=0|-) = \frac{1}{3}$.

$$\therefore P(+|A=1,B=0) = \frac{P(+,A=1,B=0)}{P(A=1,B=0)} = \frac{P(+)P(A=1|\cancel{+},B=0|+)}{P(A=1,B=0)} \sim P(A=1|+)P(B=0|+)P(+)$$

$$= \frac{3}{5} \times \frac{3}{5} \times \frac{5}{8} = 0.225$$

$P(-|A=1,B=0) \sim P(A=1|-)P(B=0|-)P(-) = 0$

by smoothing, $P(-|A=1,B=0) \sim \frac{1}{8} \times \frac{1}{3} \times \frac{1}{8} < (P+|A=1,B=0)$

$\therefore$ it will be '+' class.

IS 2020 Jinchao Li 1155133496

Q4

(ii)(a) ① $C_1 = 7$. $C_2 = 50$, $C_3 = 60$, $\therefore \begin{cases} C_1 : 7, 13, 20, 25 \\ C_2 : 30, 42, 50 \\ C_3 : 60 \end{cases}$

② update: $C_1 = 16.25$, $C_2 = \frac{1}{3}(30+42+50) = \frac{122}{3} \approx 40.7$, $C_3 = 60$

new classes distribution: $\begin{cases} C_1 : 7, 13, 20, 25 \\ C_2 : 30, 42, 50 \\ C_3 : 60 \end{cases}$

the class distribution doesn't change, break. (k-means done.)

(b)

Q5 (Information Retrieval)

(i)

① A document ~~of~~ collection

② A test suite of information needs, expressible as queries

③ A set of relevance judgements, standardly a binary assessment of re~~le~~levant or nonrelevant for each query-document pair.

(ii)

precision will decrease generally, and recall will increase until 1.

(iii) ① as recall increasing to 100% by just getting all documents, the arithmetic mean will get at least 50%, which is unsuitable.

② Moreover, F-measure is more closer to min (Precision, Recall).

(iv)

(a) let $<t_1, t_2, \cdots, t_{n_d}>$ be tokens in d. ~~$c \in C$~~ then

$$g = \arg\max_{c \in C} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k | c) \qquad \text{(or use log).}$$

(b) $\hat{P}$ is estimated by training set, in 'log' version, we can see that the predict is related to the frequency of token in the document. Multi~~mod~~nominal Model estimates $\hat{P}$ as fraction of tokens ~~and~~ or positions And Bernoulli model estimates $\hat{P}$ as fraction of documents.

(c) Multinominal Model. because Bernoulli ingores the number of occurances, and fraction of tokens/positions, which Multinominal needs them. □