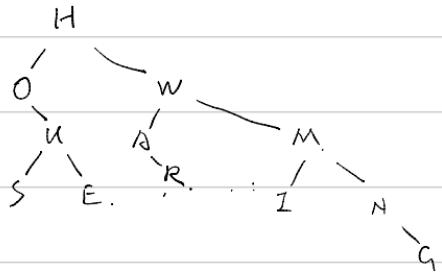




日期: / 3:20

1(a)



1(b)

2 →

2' 3 →

2' 3' 5.

→ 2' 3' 5 → 2' 3' 5' 7 → 2' 3' 5' 7' 11

→ 2' 3' 7' 11' 13 → 2' 3' 7' 11' 13' → 2' 3' 5' 7' 11' 13

→ 2' 3' 7' 11' 13' → 2' 3' 7' 11' 13' 17

→ 2' 3' 7' 11' 13' 17' 19

这题 push 和 pop 有没有
heapify 的操作啊，还是单独
插入和输出。（还是自己写
一遍稳妥）

日期：

1. (c)

Algorithm:

~~Build - Min - Heap (A):~~

~~for i ← ⌈n/2⌉ + 1 to 0~~

~~Min - heapify (A, i)~~

~~Build - Min - Heap (A):~~

~~res = list()~~

~~for val in A:~~

~~res.push(val)~~

Select - kth - Min (A, k)

return res

A ← Build - Min - Heap (A)

for i in 0 to k - 1

kth - min = A.pop()

return kth - min

Complexity : Build - Min - Heap = O(n)

Select - kth - Min = O(k · log n) + O(n)

2. (a) combine (cur, new):

cur.next = new.

new.pre = cur

return new

Merge (L, R):

if L.key < R.key:

cur = L, head = cur

lnew = L.next

rnew = R.

else:

cur = R, head = cur

lnew = L, rnew = R.next

日期: / while lnew not None or rnew not None:

if (new is None):

cur = combine (cur, rnew)

rnew = rnew.next

elif rnew is None:

cur = combine (cur, lnew)

lnew = lnew.next

else:

if (new.key < rnew.key)

cur = combine (cur, lnew)

lnew = lnew.next

else:

cur = combine (cur, rnew)

rnew = rnew.next.

return head

2b)

Check:

stack = list()

for bracket in sequence:

if bracket is "(":

stack.push("(")

else:

symbol ← stack.pop()

if symbol is ")":

raise error

日期: / if stack.size() > 0:
raise error
print("is matched")

3.(a) (ii) select r.person-id
from Review r
inner join
(select movie-id from Movie
where publisher = "KEVIN FEIGE") m
on r.movie-id = m.movie-id
group by r.person-id
having count(*) >= 2;

(ii) select Avg(m.budget)
from Movie m
inner join
(select r.movie-id
from Role r
inner join
(select person-id
from Person
where name = "CHRISTIAAN BALE") p
on r.person-id = p.person-id) r2
on m.movie-id = r2.movie-id

3.(b) (ii)

M1 ← $\cap_{Movie.id} \{ \text{publisher} \neq "KEVIN FEIGE" \}$ (Movie)

$\Pi_{person.id}(Review) - \Pi_{person.id}(Review \cap M1)$

日期:

(ii) $M2 \leftarrow \Pi_{\text{movie-id}} \delta_{\text{publisher} = "KEVIN FEIGE"} (\text{Movie})$

$R_watch \leftarrow \Pi_{\text{person-id}, \text{movie-id}} (\text{Review}) \bowtie M2$

$R_unwatch \leftarrow \Pi_{\text{person-id}} (\text{Review}) \times M2 - R_watch$

$P \leftarrow \Pi_{\text{person-id}} (\text{Review}) - \Pi_{\text{person-id}} (R_unwatch)$

$\Pi_{\text{name}} (P \bowtie \Pi_{\text{person-id}, \text{name}} (\text{Person}))$

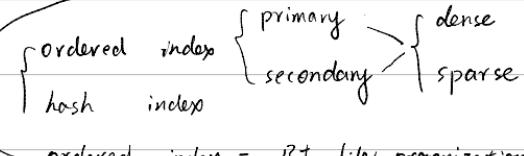
3.(c) B^+ file organization?



Role save as

dense primary index on (role) attribute,

(movie-id, person-id) as secondary index



ordered index = B^+ file organization

hash = hash file organization

Person \swarrow B^+ file organization?

save as

dense primary index on (name) attribute.

secondary index on (person-id) attribute.

Review \swarrow hash file organization?

save as

hash index, compute hash values on (person-id, movie-id).

日期: /

4(a) Δ doctor: (doctor-id, name, room-number, gender)

Δ user: (user-id, name, gender, date-of-birth)

Δ email: | email-address, user-id)

since email is multi-value attribute, it need
2 attributes as primary key.

Δ appointment: (doctor-id, user-id, appointment-datetime)

since appointment is (many-to-many) relationship,
it need 2 attributes as primary key.

日期:

4(b) ① 1 attribute :

$$A^+ = (A, C, E) \quad \text{not candidate key.}$$

$$B^+ = (B, D) \quad \text{no}$$

$$C^+ = (C), \quad D^+ = (D), \quad E^+ = (E) \quad \text{no}$$

② 2 attributes :

$$(AB)^+ = (A, B, C, E, D), \quad \text{yes}$$

$$(CD)^+ = (AB \cup DE), \quad \text{yes}$$

$$(AD)^+ = (AB \cup DE), \quad \text{yes}$$

$$(BC)^+ = (AB \cup DE), \quad \text{yes}$$

③ 3 attribute :

if attribute set doesn't contain A,

it must contain {CD}, so it can infer to A.

but {CD} is candidate key.

$$\text{consider } \{ACE\}^+ = (ACE), \quad \text{no}$$

In conclusion (AB), (CD), (AD), (BC).

日期:

q(4) check: $A \rightarrow C$, it's not trivial, A is not superkey of R

so R is not BCNF.

decompose: ① consider R:

$A \rightarrow C$ not trivial, $A \cap C = \emptyset$,

A is not superkey

$$\text{set} = \{(A, B, D, E), (AC)\}$$

② consider (ABDE)

$A \rightarrow E$ not trivial, $A \cap E = \emptyset$,

A is not super key.

$$\text{set} = \{(ABD), (AE), (AC)\}$$

③ consider (ABD)

$B \rightarrow D$ not trivial, $B \cap D = \emptyset$

B is not super key.

$$\text{set} = \{(AB), (BD), (AE), (AC)\}$$

$$\text{BCNF} = \{(AB), (BD), (AE), (AC)\}$$