Q1/

a) Since we are searching data on ranged query only, it is possible to evaluate it with an index-only plan.

b) It is not possible since to answer the query, we must access data record too.

Q2/

a) Since we are searching on equality condition, hash index is the cheapest.

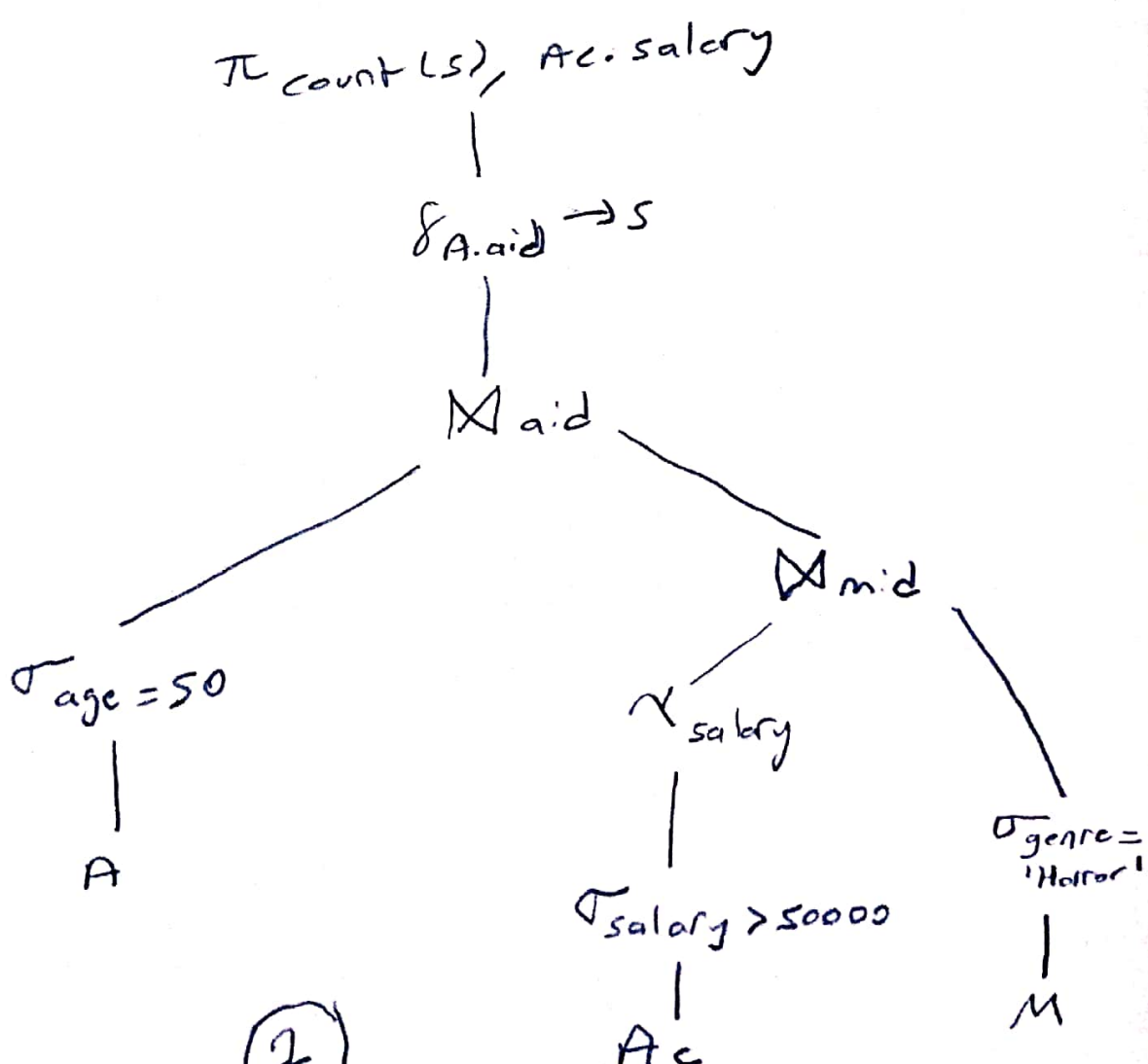
b) Clustered B+ tree is the better one because we are interested in range condition with 19.999 records, which is greater than number of records that pages can contain.

c) The best choice is unclustered B+ tree because we need to find 9 records with ranged query, but every page contains 10 records, which means clustering makes the running query more slower.

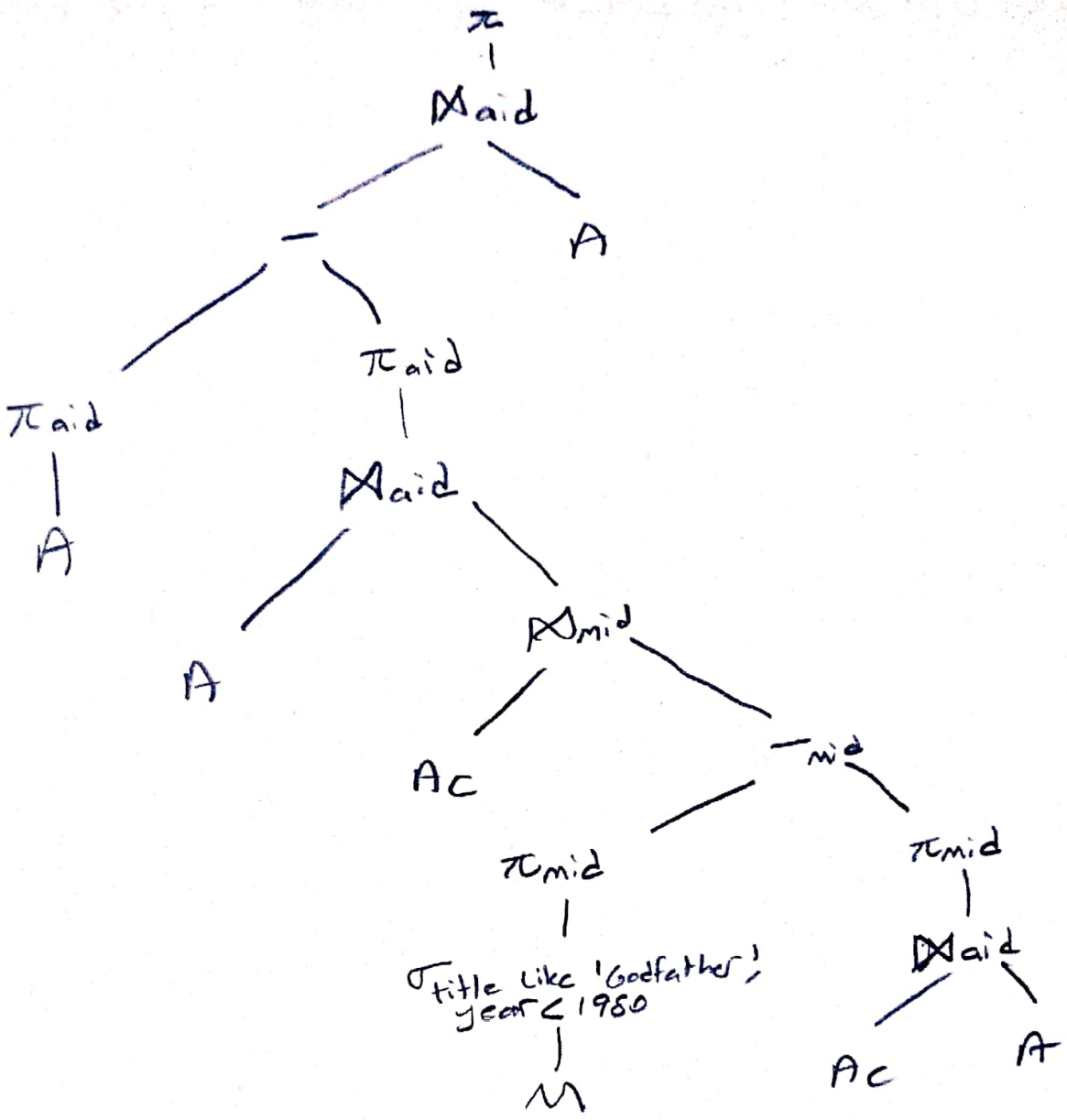
d) Since we must reduce the disk access cost, using heap file is better solution.

Q3/

a)



b)



Q4

a) $B(R) = 2000$
$B(S) = 500$
$M = 42$

$$\text{Cost} = 2000 + \frac{2000 \cdot 500}{40} = \boxed{27000}$$

b) $$\text{Cost} = 500 + \frac{(B(R) + \frac{B(R) \cdot B(S)}{M-2}) \cdot 500}{40} = \boxed{25500}$$

$$\left(B(S) + \frac{B(R)\,B(S)}{M-2}\right)$$

c)

d) Since $B(R) > M$ and $\min(B(R), B(S)) = 500 < M^2$, we need to use partitioned hash-join.

In partitioning, we are using all of pages of memory (one for input buffer, $M-1$ pages for buckets). when one of bucket is full, it is written to disk.

In joining, again we create another hash table in memory and join blocks in disk. We use all of pages in memory.

Cost: $3(2000 + 500) = \boxed{7500}$

$(3(B(R) + B(S)))$

e) If S has an index or join attribute, it iterates over R, for each tuple fetch corresponding tuple from S.

i) $V(S,b) = 5000$, $T(R) = 20000$, $T(S) = 5000$

Cost: $2000 + \frac{20000 \cdot 500}{5000} = 4000$

$\left(\text{Cost} = B(R) + \frac{T(R)B(S)}{V(S,b)}\right)$

ii) cost: $B(R) + \left(\frac{T(R) \cdot T(S)}{V(S,b)}\right)$

$= 2000 + \frac{20000 \cdot 5000}{5000} = 22000$

Note that $V(S,a) = 5000$ because in joining $R \underset{R.a}{\bowtie} \underset{S.b}{S}$, S.b is the primary key.

Q5/

a) $(m10 + m11 + m12) * \left(\frac{t_{swim}}{N}\right)$

b) We assume that the data is uniformly distributed among the type and month attributes, and data exist in all attribute values.

This estimate can be incorrect if data is not distributed uniformly or data is not exist in any value of attribute.