4. Query Execution [25 points]
You have been hired to advise a major hot-dog vending franchise called “Dunce Dog” to tune their database server installation. They run a commercial DBMS called Tentacle version 9y. One of the startup parameters in Tentacle is called `query_space`; it tells Tentacle how many disk-blocks worth of RAM it should allocate to use for sorting and hash joins. Note that memory used for `query_space` is separate from the Tentacle buffer pool.

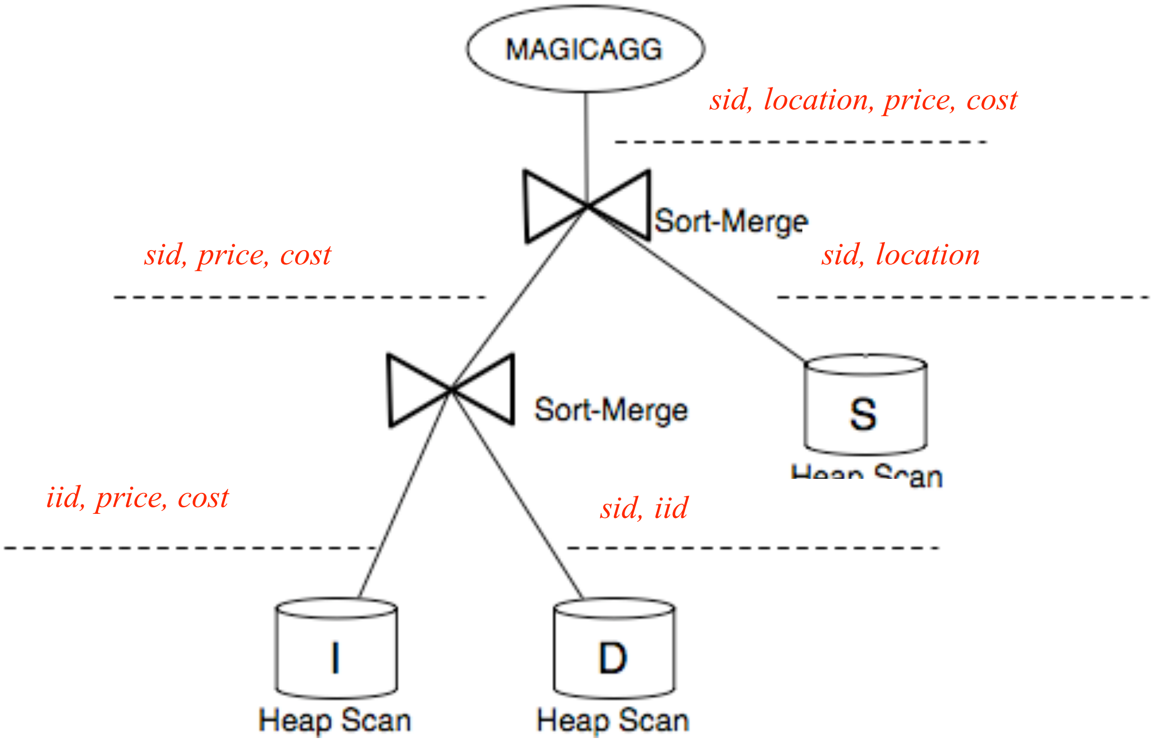
Dunce Dog is having problems running one of their monthly reporting queries. It uses the following tables:

Store(<u>sid</u>, location, owner)
ItemsForSale(<u>iid,</u> name, description, cost, price)
DetailedSales(<u>receiptno</u>, <u>iid</u>, sid)

The query is:

```
SELECT S.sid, S.location, SUM(I.price - I.cost) AS PROFIT
  FROM Stores S, ItemsForSale I, DetailedSales D
 WHERE S.sid = D.sid and D.iid = I.iid
GROUP BY S.sid, S.location
```

You determine that the query plan that Tentacle chooses is:



Name____

a) *(5 points)* Tentacle performs aggressive projection – that is, it discards any attributes that it does not need as soon as possible. For each edge in the query plan, write down the smallest list of attributes that needs to be *retained* (i.e. not discarded) on the corresponding dotted line.

b) *(5 points)* Assume that the result of the scan and projection of ItemsForSale fits in 1000 pages, and the result of the scan and projection of DetailedSales fits in 10,000 pages. Approximately how many blocks of `query_space` should Tentacle need *at minimum* to perform the sort-merge join of ItemsForSale and DetailedSales in two passes? Feel free to round up or down by as many as 2 blocks in any equations that you use, but if you do so, show your equations!

sqrt(MAX(10000 , 1000)) = 100 blocks

c) *(5 points)* Assume that the result of the first join fits in 2,500 pages, and the result of the scan of Stores fits in 400 pages. Approximating and showing work as in part (b), estimate the number of blocks of `query_space` that Tentacle should need to perform the second join in the plan in two passes.

sqrt(MAX(2500, 400)) = 50 blocks

d) *(5 points)* How much memory should Tentacle's MAGICAGG operator need to perform GROUP BY and aggregation without I/O? Again, feel free to round up or down by as many as 2 blocks.

0 blocks, because of the "interesting order" resulting from the sort-merge join (and the fact that sid is a key means one location per sid).

e) *(5 points)* Using the variables *b, c* and *d* to represent your answers to parts (b), (c) and (d) respectively, which of the following represents the minimum value of `query_space` that your client should use:

a. SUM(b, c, d)

b. MAX(b, c, d)

c. MIN(sqrt(b), sqrt(c), d)

d. $\log_{\text{query\_space}}$ MAX(b, c, d)

YOUR ANSWER HERE (4e): ________*b*___________

Name____

EXTRA CREDIT: The Disk Whisperer (10 points)

It's exhausting being a disk drive, constantly seeking, scanning, transferring… Since you are a sensitive Berkeley person, you are able to talk to your disk drive and it confides in you about how tough life is.

Recently, your disk drive told you that it's really tired of doing sort-merge joins. You offered to let it do hash joins all the time, even when the optimizer used to choose sort-merge join. Your disk responded, "Who cares – same stuff, different order."

Explain why your disk drive said that.

From the disk's perspective, these algorithms do the same physical operations: the same numbers of seeks, and the same amount of scanning, and transferring.

Sort-Merge: Pass 0 is all sequential I/O on each relation, so 1 seek and N block scan/transfer operations to read each relation, 1 seek and N block scan/transfer operations per relation to write. Pass 1 is random I/O to refill buffers from the various runs as they are drained, so N seeks and N block scan/transfer operations per relation.

Hash-Join: Phase 1 is sequential reads but random writes to hash partitions. So 1 seek and N block scan/transfer operations to read each relation, N seeks and N block scan/transfer operations per relation to write the partitions. Phase 2 is sequential reads to construct a hash table on one relation (1 seek and N scan/transfer ops) and sequential read to stream the other relation in and probe the hah table (1 seek and N scan/transfer ops.)