Homework 3

1. (60 points) Consider the join of R and S where *R.a = S.b*, given the following information about the relations to be joined. The cost metric is the number of disk I/Os and the cost of writing out the result is ignored.
   - Relation *R* contains 10,000 tuples; each tuple is 400 bytes long.
   - Relation *S* contains 2,000 tuples; each tuple is 400 bytes long.
   - A page size is 4096 bytes and the unpacked, bitmap page format is used. For each page, 96 bytes are reserved and cannot be used to store data. The rest of the page is used to store as many tuples as possible.
   - Attribute *b* of relation *S* is the primary key for *S*.
   - Both relations are stored as simple heap files. Neither relation has any index built on it.
   - The available memory is 52 pages.
   - The fudge factor is 1.1.

   Answer the following questions and explain why:

   a) What is the cheapest cost of joining *R* and *S* using a block nested loops join for the given amount of memory buyer space? What should the number of buffer pages be so that the cost of the join is the minimum?
   b) What is the cheapest cost of joining *R* and *S* using a GRACE hash join?
   c) What is the cheapest cost of joining *R* and *S* using a sort-merge join?

Each page have 4000 usable bytes.

Each page of R contain 10 tuples. R takes 1,000 pages. S takes 200 pages

a)M+ceiling(M/(B-2))*N

$M + \lceil \frac{M}{B-2} \rceil * N$ S

R as outer

1000+1000/50*200=1000+4000

S as outer

200+200/50*1000=200+4000=4200 cheapest

So cost of join is minimum if one relation can be put in buffer entirely. At least 200+2=202

$\frac{200}{202-2} = 1$

b) validate if partition of can fit into memory. 52>sqrt(1.1*200), OK. We can partition on S. 3*(M+N)=3600

c) sorting should be multi-way merge sort.

We can have R sorted in 2 passes. Read and Write in each pass. 2*2*1000= 4000

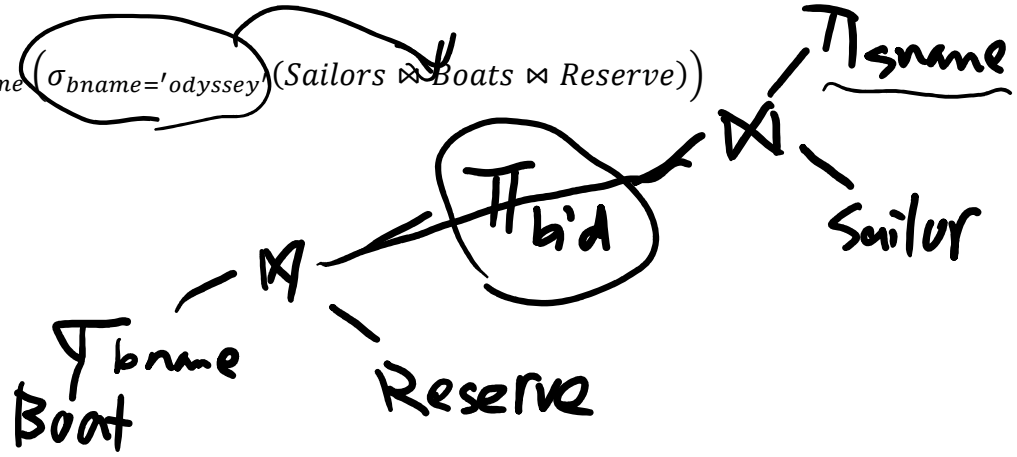
We can have S sorted in 2 passes as well. 2*2*200=800

Join: 200+1000. Total:4000+800+200+1000=6000

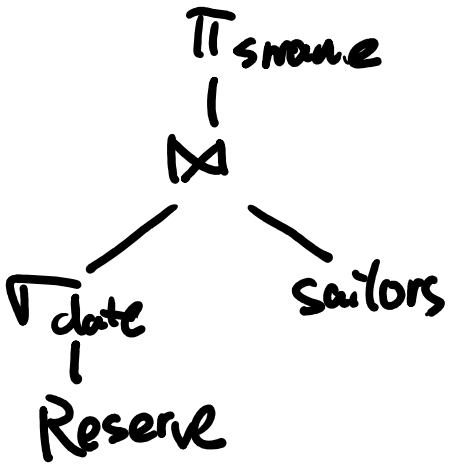
2. (40 points) Consider three relations *Sailors*(*sid, sname, rating, age*), *Boats*(*bid, bname, color*), and *Reserve*(*sid, bid, day*) and the following queries
    - Find the names of sailors who reserved boat "*Odyssey*"
    - Find the names of sailors who reserved at least one boat on day "*05/15/2020*"
    - Find the names of sailors who have not reserved any boat

    a) Write the relational algebraic expressions for each of the above queries.
    b) Draw their expression trees. Your trees do not have to be optimized, but make sure that 1) selection and projects are done as early as possible, and 2) left-deep joins are used whenever joins are needed.

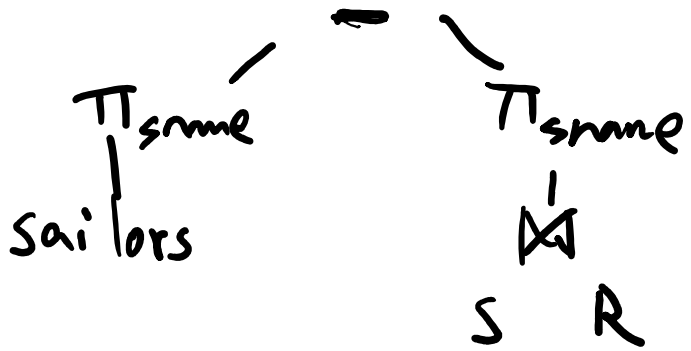
a) $\pi_{sname}\left(\sigma_{bname='odyssey'}(Sailors \bowtie Boats \bowtie Reserve)\right)$



2) $\pi_{sname}\left(\sigma_{day\ '05\backslash 15\backslash 2016'}(Reserve) \bowtie Sailors\right)$



3) $\pi_{sname}(Sailors) - \pi_{sname}(Reserve \bowtie Sailors)$



Submission Instruction

Please use Microsoft Words or other tools to type your answer. Don't handwrite. Submit your answer in pdf format on Canvas.