Comp115 Spring 2017, HW4 (bonus)

Due April 11th, 2017

Submit via provide using:

```
$ provide comp115 115HW4-QEval <single_file.pdf>
```
or online at: `https://www.cs.tufts.edu/comp/115/provide.cgi`

Consider the join R$\bowtie_{R.a=S.b}$S, given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted. The cost of writing out the result should be uniformly ignored, but bear in mind that one output buffer should be used for producing the result.

Relation R contains 200,000 tuples and has 20 tuples per page.
Relation S contains 4,000,000 tuples and also has 20 tuples per page.
Attribute a of relation R is the primary key for R.
Each tuple of R joins with exactly 20 tuples of S.
Both relations are stored as simple heap files.
Neither relation has any indexes built on it.
1002 buffer pages are available.

1. What is the cost of joining R and S using a page-oriented simple nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged? **[10 pts]**
2. What is the cost of joining R and S using a block nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged? **[10 pts]**
3. What is the cost of joining R and S using a sort-merge join? What is the minimum number of buffer pages required for this cost to remain unchanged? **[20 pts]**
4. What is the cost of joining R and S using a hash join? What is the minimum number of buffer pages required for this cost to remain unchanged? **[20 pts]**
5. What would be the lowest possible I/O cost for joining R and S using any join algorithm, and how much buffer space would be needed to achieve this cost? Explain briefly. **[15 pts]**
6. How many tuples does the join of R and S produce, at most, and how many pages are required to store the result of the join back on disk? **[15 pts]**
7. Would your answers to any of the previous questions in this exercise change if you were told that S.b is the primary key for S? **[10 pts]**