Exercises for 3/12/13

1. Give one-pass algorithms, and specify when they can be used, for:
   (a) Set union
   (b) Set intersection
   (c) Bag intersection
   (d) Bag union
   (e) Outerjoin

2. For each of the operations below, write an iterator for a one-pass algorithm. Is the operation *blocking*, i.e., must the whole relation be read before the first output tuple is produced?
   (a) Projection
   (b) δ
   (c) γ
   (d) Set union
   (e) Bag intersection

3. Nested-loop join
   (a) Write iterator methods for the block-based version of nested-loop join
   (b) $B(R) = B(S) = 10.000$, $M = 1001$. Calculate the I/O cost of a nested-loop join
   (c) For the same relations, what value of M is needed to compute $R \bowtie S$ with nested-loop join and no more than (a) 100.000 (b) 25.000 and (c) 15.000 disk I/O s?

4. Describe 2-pass sort-based algorithms for the following. If the relation(s) have 10.000 blocks each, how much memory is needed?
   (a) γ
   (b) Set intersection
   (c) Set union

5. Suppose that there is an index on $R.a$. How can the index be used to improve the execution of the following operations? When is it more efficient than sort- or hash-based algorithms?

   (a) $R \cup_S S$

   (b) $R \cap_S S$

   (c) $\delta(R)$

6. Let$B(R) = 10.000$ and $T(R) = 500.000$ with an index on $R.a$ and $V(R, a) = k$. Give the I/O cost of $\sigma_{a=0}(R)$ as a function of k in the case that (a) we use the index and (b) we don't use it

7. For each of the following schedules,

   - Construct the precedence graph
   - Is the schedule conflict-serializable?
   - If so, what are all the equivalent serial schedules?

   (a) $r_1(A); r_2(A); r_3(B); w_1(A); r_2(C); r_2(B); w_2(B); W_1(C)$

   (b) $r_1(A); w_1(B); r_2(B) : w_2(C); r_3(C); w3(A)$

   (c) $w_3(A); r_1(A); w_1(B); r_2(B) : w_2(C); r_3(C);$

   (d) $r_1(A); r_2(A); w_1(B); w_2(B); r_1(B); r_2(B); w_2(C); w_1(D)$

   (e) $r_1(A); r_2(A); r_1(B); r_2(B); r_3(A); r_4(B); w_1(A); w_2(B)$