

## CS386D Problem Set #2

Due Monday, September 29th

- [1] How can the following operations on tuple streams be parallelized?
- (a) Maximum — find the maximum-valued element in a stream of numbers.
  - (b) LongestSequence — return the length of the longest sequence of identical consecutive tuples in a stream.
- [2] Justify the statement: “as long as the amount of memory available is between the square root of the data file size and the size of the entire data file, the file can always be sorted into two passes”.
- [3] Given the following SQL select statement:

```
select *
from R
where A.exceeds(20) and B.includes(40) and C.overlaps(40,20)
```

- (a) Suppose the exceeds predicate has an execution cost of 5 and a selectivity of .3, the includes predicate has an execution cost of 20 with a selectivity of .1; and the overlaps predicate has an execution cost of 7 and has a selectivity of .2. What order of evaluation minimizes query evaluation costs?
  - (b) Now suppose the costs for each of these predicates are 0 (or really something very small). What order of evaluation would minimize query evaluation costs?
- [4] Suppose an R-tree index is over attribute A, which is of type rectangle. Attribute B is not indexed. Consider the following select:

```
select *
from R
where A.overlaps(rectangle(1,3,4,6)) and B=17;
```

If the cost of evaluating the overlaps predicate is 5 with a selectivity of .1 and the cost of evaluating the equals predicate is 0 with a selectivity of .4, how should this query be processed so that execution costs are minimized? Assume R has a large number of tuples.