

4. Übungsblatt

Ausgabe: 14. November 2013 · Besprechung: 20. November 2013

1 Block Nested Loops Join vs. Index Nested Loops

Gegeben seien zwei Relationen R und S , wobei $|R| = 1000$ Seiten und $|S| = 100\,000$ Seiten. Jede Seite enthält 20 Tupel. Beide Relationen befinden sich auf einer Festplatte mit einer Zugriffszeit von 10 ms und einer Lesegeschwindigkeit von 10000 Seiten/Sekunde.

1. Wie lange benötigt ein **Block Nested Loops Join**-Algorithmus mit einer Blockgröße von 100 Seiten, um $R \bowtie S$ zu berechnen?
2. Nehmen Sie an, es existiert ein Index auf dem Joinattribut von Relation S . Der Index liege vollständig im Cache (Buffer Manager), ist jedoch nicht geclustert. Der zu berechnende Join $R \bowtie S$ habe eine Selektivität von 1 %, d. h., jedes Tupel aus R passt zu 1 % aller Tupel aus S . Wie lange benötigt ein **Index Nested Loops Join**, um $R \bowtie S$ zu berechnen?

2 Hashing vs. Sorting

Die Ansätze Hashing und Sortieren haben überraschend viele Gemeinsamkeiten (wir haben z. B. in der Vorlesung bereits gesehen, dass Sort-Merge Join und Hash Join ähnlichen Aufwand haben).

Wie lassen sich folgende Aufgaben mittels Hashing bzw. mittels Sortieren effizient durchführen? Skizzieren Sie jeweils einen Algorithmus als Pseudocode.

1. Duplikatelimination (“**SELECT DISTINCT**”)
2. Ein **UNION** mit Duplikatelimination
3. Eine Berechnung der Schnittmenge (“**INTERSECT**”)