

COMP9315 Sample Exam, Q7 Sample Solution

(a) Simple hash join ($R(id,x,y)$, $S(rid,a,b,c)$, $bR = 100$, $bS = 500$, $B = 10$)

1 input buffer for R, 1 input buffer for S,
7 buffers for in-memory hash table, 1 output buffer

```

clear hash table
for each page pR in R {
    read page pR
    for each tuple tR in page pR {
        hR = hash(tR)
        if (buffer[hR] is full) {
            for each page pS in S {
                read page pS
                for each tuple tS in page pS {
                    hS = hash(tS)
                    check for joins with tS in buffer[hS]
                }
            }
        }
        clear hash table
    }
    add tR to buffer[hR]
}

```

Requires multiple passes over S, but only one pass over R
How many passes over S? If we assume completely uniform distribution of key values in R, the hash table will fill approximately $bR/7$ times.

Cost = $bR + \text{ceil}(bR/7) * bS = 100 + 15 * 500 = 7600$ reads

(b) Hybrid hash join ($R(id,x,y)$, $S(rid,a,b,c)$, $bR = 100$, $bS = 500$, $B = 20$, $k = 1$)

Phase 1:

1 input buffer, 10 buffers for in-memory hash table, 9 buffers for partitioning

Ooops ... I forgot about the output buffer that we need in the second part of this phase ... if we can't assume that any bucket of R fits in less than 10 pages, then we need to assume 21 buffers, and thereby get the required output buffer.

```

for each page in R {

```

```
    for each tuple in page {
        h = hash(t)
        if (h == 0)
            add to in memory hash table
        else
            add to disk partition h
    }
}
for each page of S {
    for each tuple in page {
        h = hash(t)
        if (h == 0)
            perform join via in memory hash table
        else
            add to disk partition h
    }
}
```

Phase 2:

1 input buffer, 18 buffers for in-memory hash table, 1 buffer for output

```
for partitions p in 1..9 of R {
    read partition into in-memory hash table
    for each page in partition p of S {
        for each tuple t in page {
            perform join via in-memory hash table
        }
    }
}
```

Cost:

Phase 1: b_R reads + b_S reads + $0.9 \cdot b_R$ writes, + $0.9 \cdot b_S$ writes

Phase 2: $0.9 \cdot b_R$ reads + $0.9 \cdot b_S$ reads + X writes (depends on # results)

Cost = $100 + 500 + 90 + 450 + 90 + 450 = 1680$