

Modified Figure 19.3 (a)

=====

Implementation of the sort-merge join between tables R and S with (R.A = S.B)

```
sort the tuples in R on attribute A; // assume R has n tuples (records)
sort the tuples in S on attribute B; // assume S has m tuples (records)

i = 1; // initialize the record pointer of table R
j = 1; // initialize the record pointer of table S

while ((i <= n) && (j <= m)) {
    if (R[i].A > S[j].B) {
        j++; // advance the record pointer of S;
    }

    elseif (R[i].A < S[j].B ) {
        i++; // advance the record pointer of R
    }

    else { // R[i].A == S[j].B, so we output all matched pairs of tuples

        p = i; // p is the auxillary record pointer of table R
        while ((p <= n) && (R[p].A == S[j].B)) {

            q = j; // q is the auxillary record pointer of table S
            while ((q <= m) && (R[p].A == S[q].B)) {
                output the combined tuple <R[p],S[q]> to T; //T is the result table
                q++;
            }

            p++;
        }

        i = p; // update the primary record pointer of table R
        j = q; // update the primary record pointer of table S
    }
}
```