Overview

```
SELECT A.a1, B.a2, B.a3, D.a4
FROM A
JOIN B ON A.id = B.a_id
JOIN C ON B.id = C.b_id
WHERE A.a1 = 42
AND B.a3 = 12
```
The Story so far:

```
SELECT A.a1, B.a2, B.a3, D.a4
FROM A
  JOIN B ON A.id=B.a_id
  JOIN C ON B.id=C.b_id
WHERE A.a1=42
    AND B.a3=12
```
How to Proceed?

iseek(a1=42, A) -> SHJ -> MJ_{b.id=B.id} -> sort_{b.id} -> scan(C) -> filter_{a3=12} -> isam(id, B)

how to execute?
Approach 1: Function Library

\[
\text{SHJ(} \text{iseek(a1}=42, \ A) , \ \text{MJ}_C.b\_id=B.id( \ \text{sort}\_b\_id( \ \text{scan}(C) ), \ \text{filter}\_a3=12( \ \text{isam}(id, B) ) ) )
\]

how to execute?
Problem: Intermediate Results

- $\text{iseek}(a1=42, A)$
- $\text{sort}_{b\_id}(\cdot)$
- $\text{filter}_{a3=12}(\cdot)$
- $\text{scan}(C)$
- $\text{isam}(\text{id}, B)$
Approach 2: Pipelining

```

uni:x/linux

grep < stdin > | grep ... > out.txt
```
Approach 2: Pipelining

iseek(a1=42, A) -> sort_{b_id} -> MJ_{C.b_id=B.id} -> filter_{a3=12} -> scan(C) -> isam(id, B)
Stages of SHJ/INLJ

1) reading

2) left input exhausted

3) Looping and Probing

(A) just buffer

(B) fully-inset

(A) bulkload index

(C) done

(A + B) index probe
Stages of Quick-Sorting

(1) reading

(2) sorting

(3) outputting
Stages of External Merge Sort

1. Reading (with run generation)
2. Merging (without final merge)
3. Outputting (with final merge)
Blocking vs Non-Blocking Algorithms

INLJ/SHJ (with bulkload) vs INLJ/SHJ (without bulkload)

bulkload

filter

project

INLJ/SHJ (with bulkload) = bulkload + INLJ/SHJ (without bulkload)