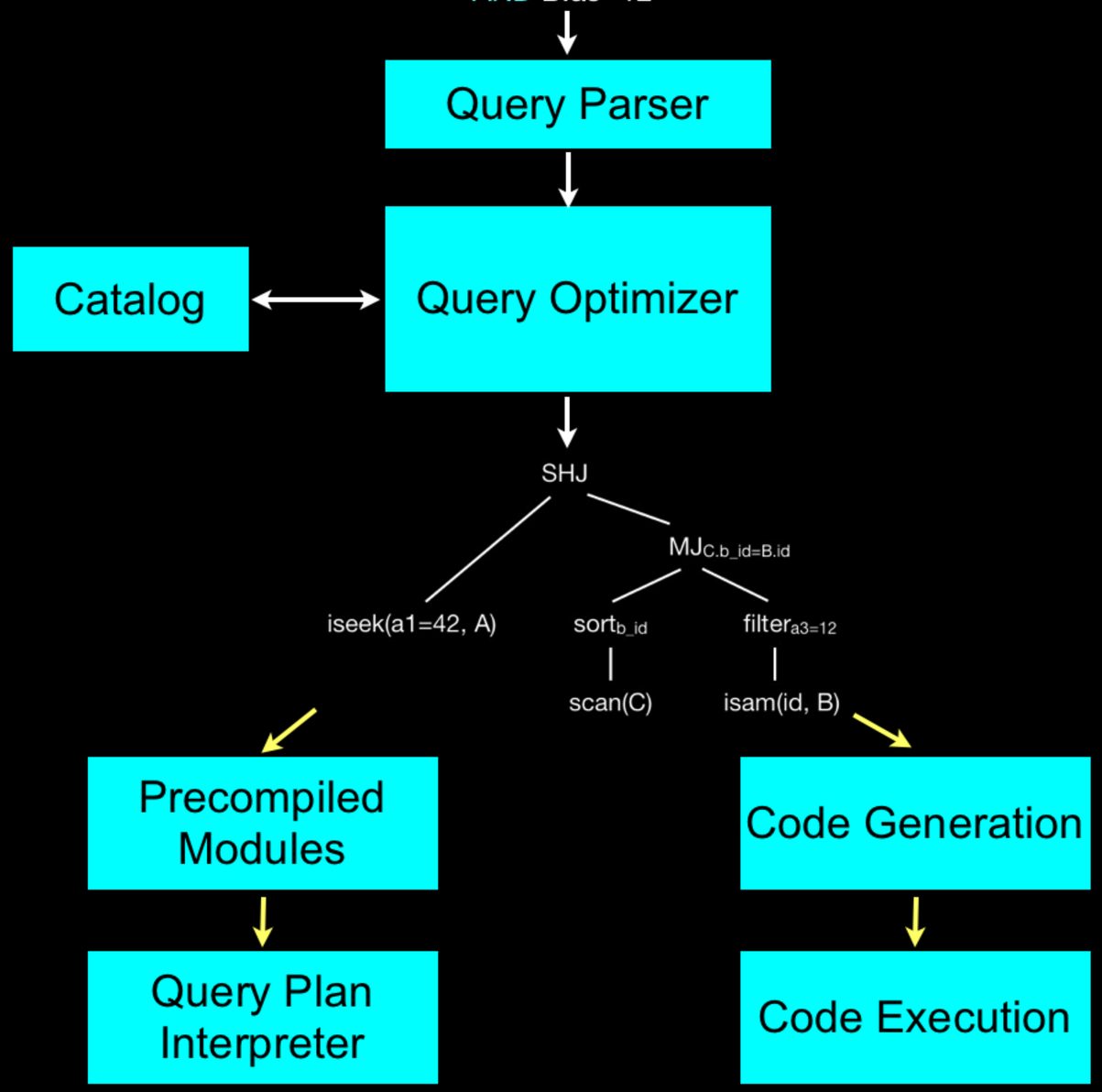
Overview

```
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
```

declarative query



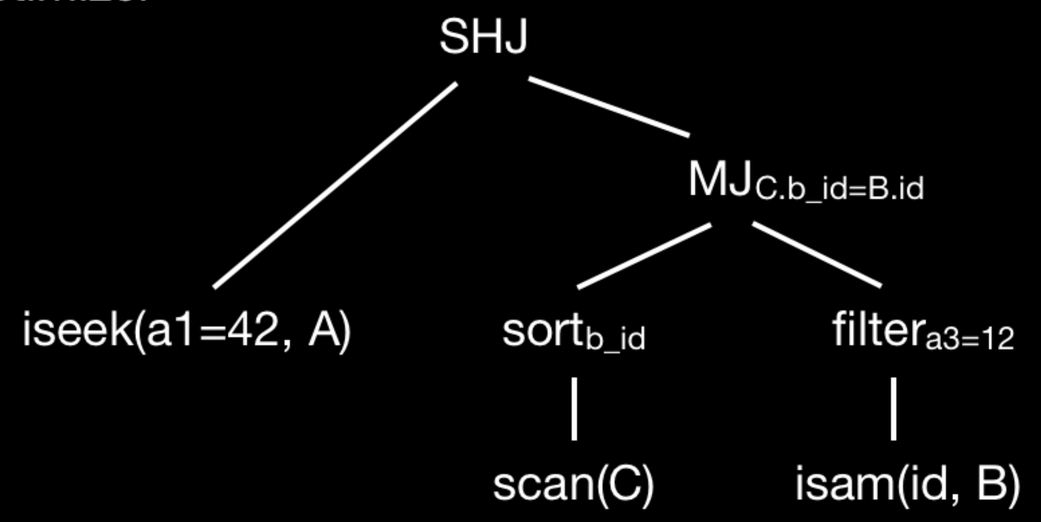
physical query execution plan (QEP)

how to execute?

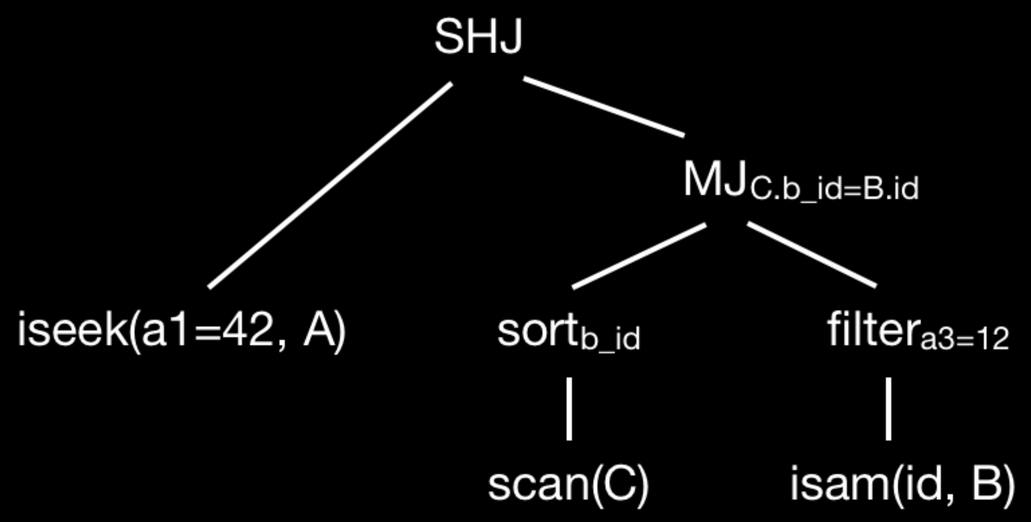
The Story so far:

```
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
```

Query Parser/Optimizer

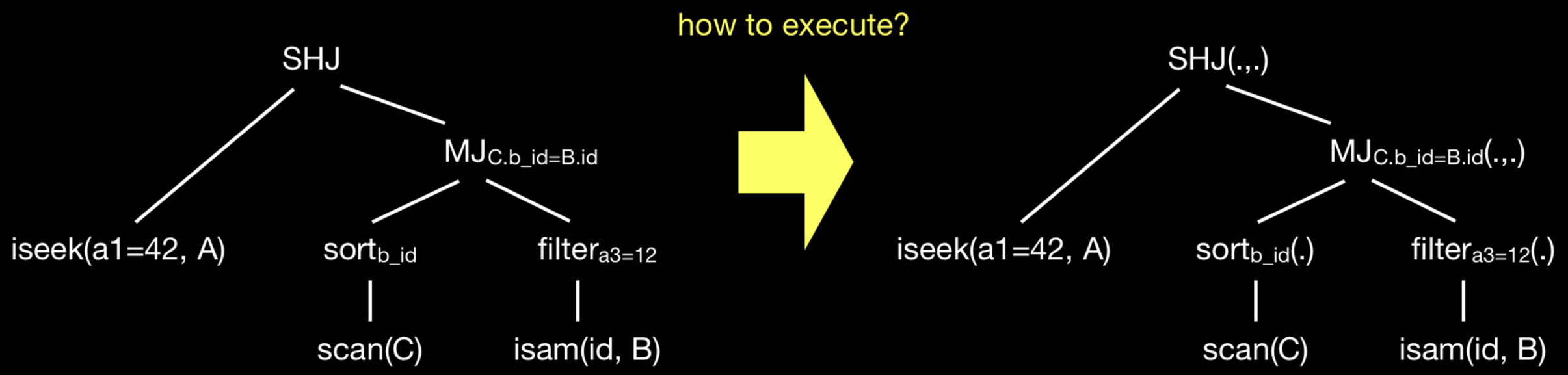


How to Proceed?



how to execute?

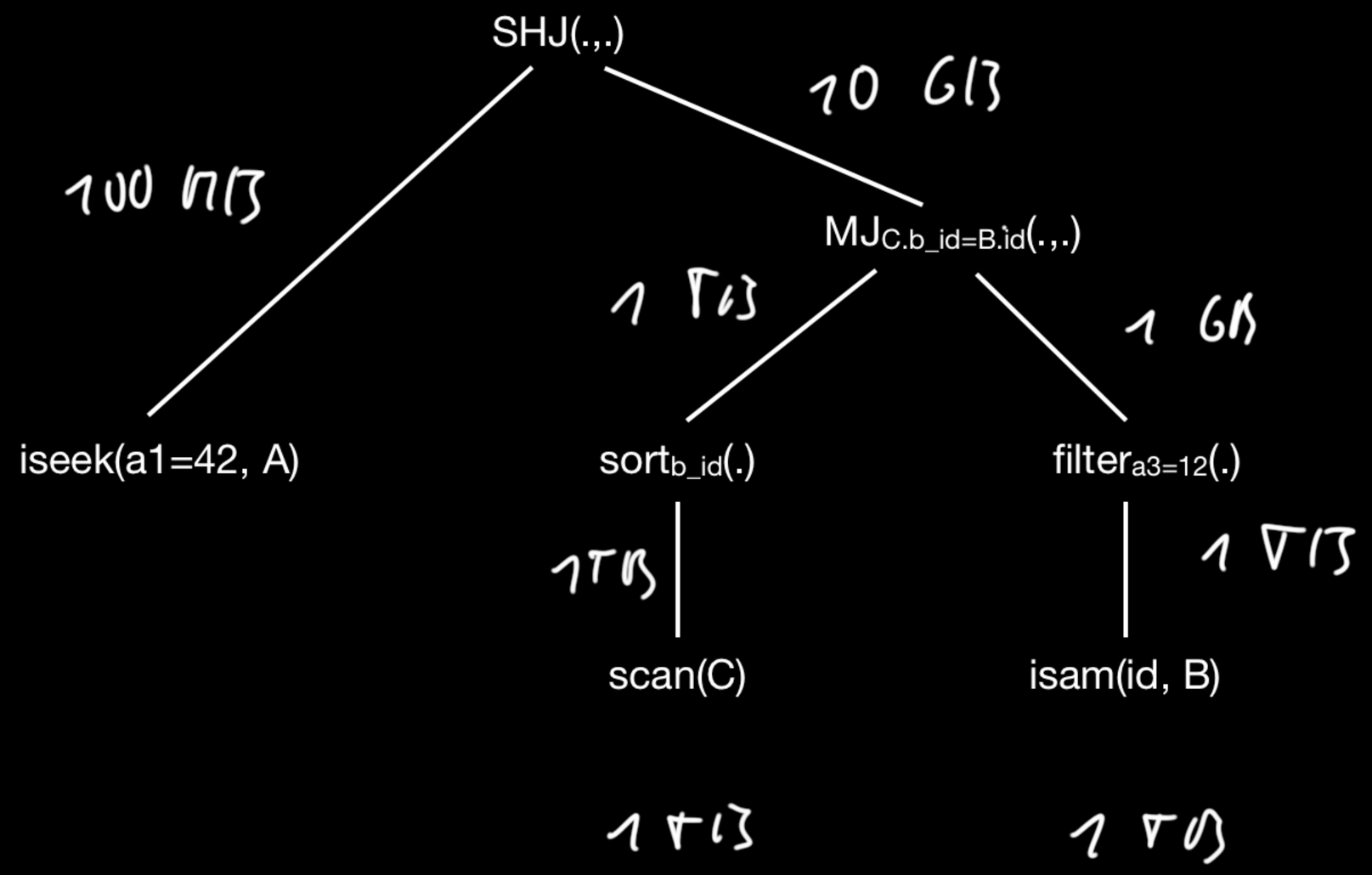
Approach 1: Function Library



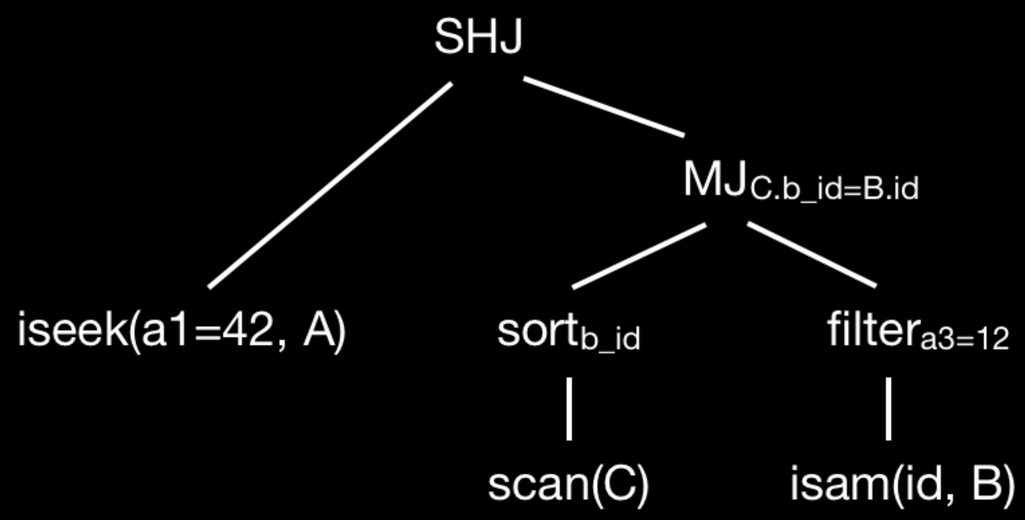
||

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

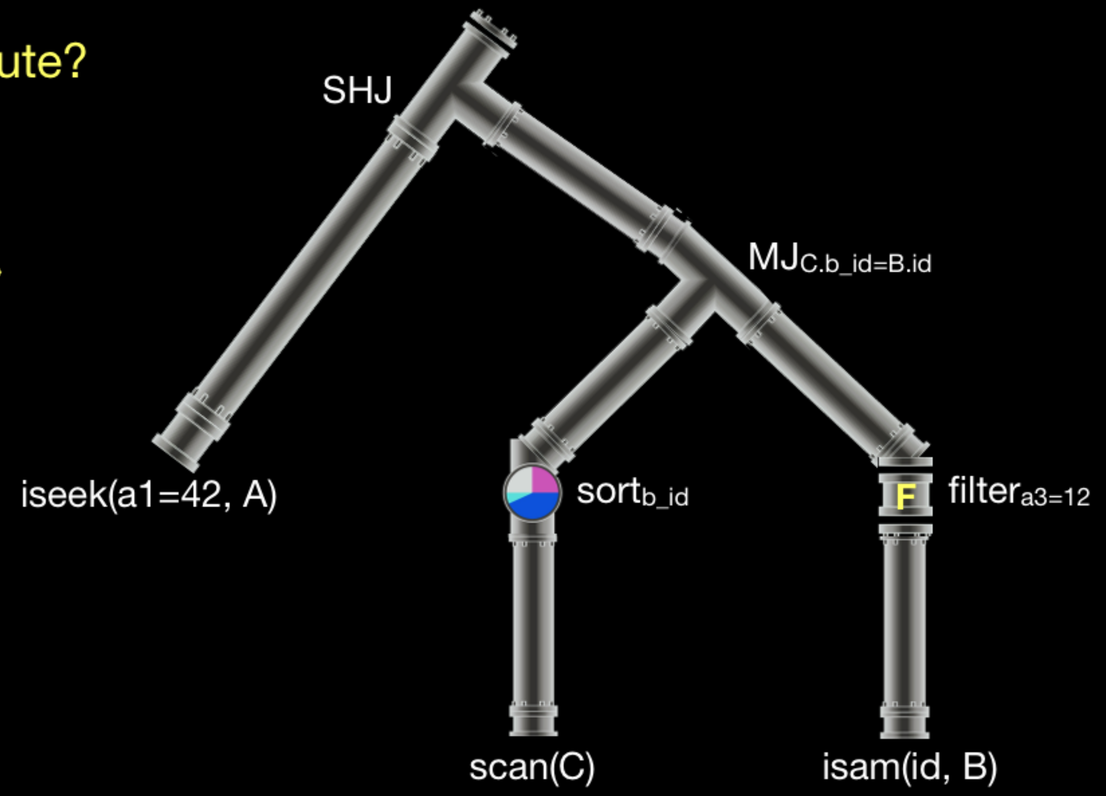
Problem: Intermediate Results



Approach 2: Pipelining



how to execute?

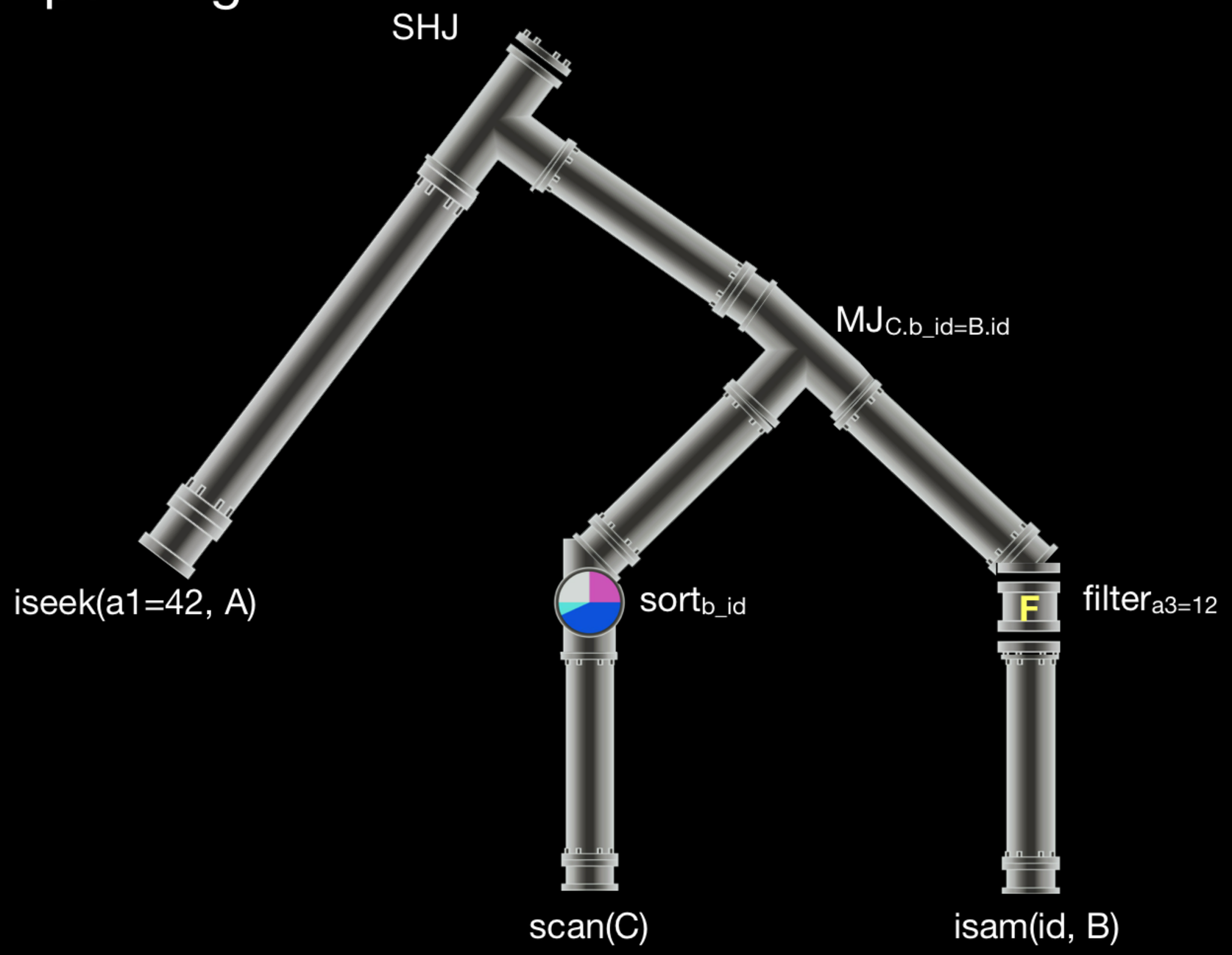


unix/linux

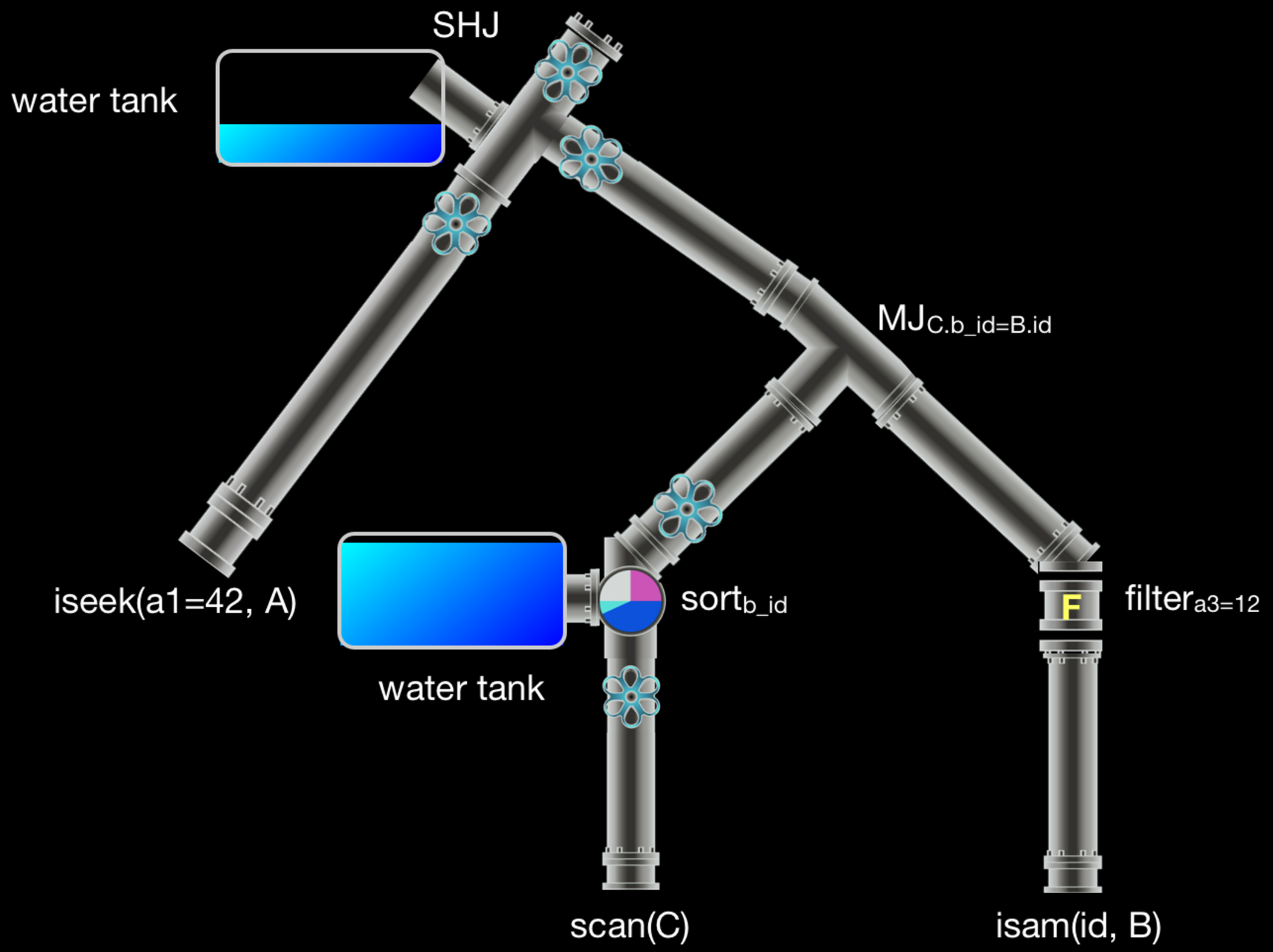
grep <string> | grep ... > out.txt

↑ pipe symbol

Approach 2: Pipelining



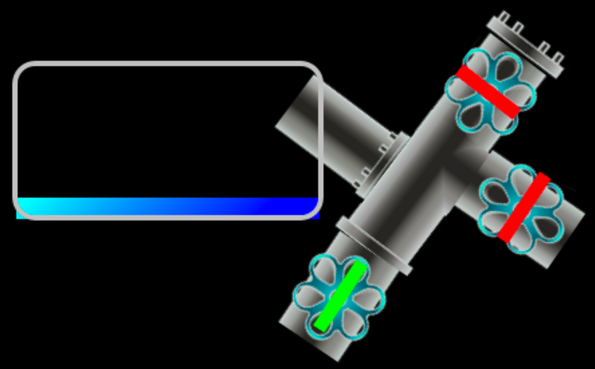
Pipeline Breakers?



Stages of SHJ/INLJ

blocking

(1) reading

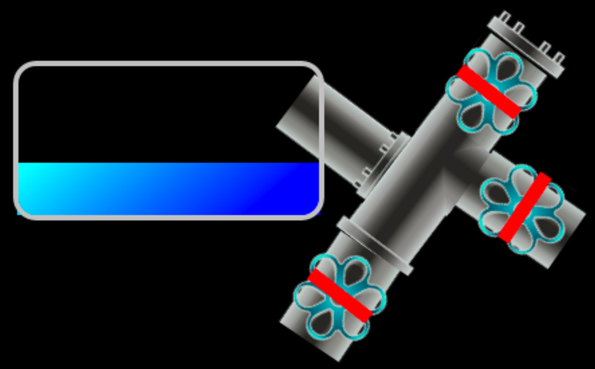


(A) just buffer

(B) tuple-wise insert

blocking

(2) left input exhausted

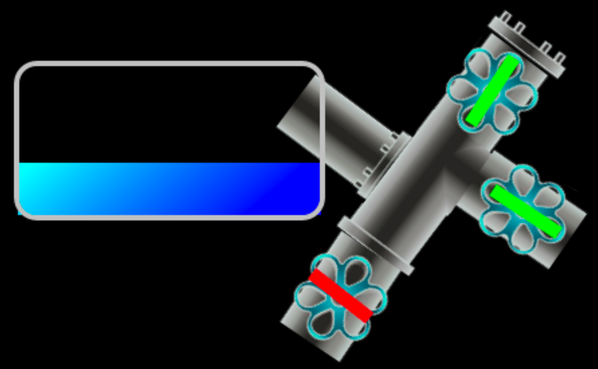


(A) bulkload index

(B) Done.

non-blocking

(3) Looping and Probing

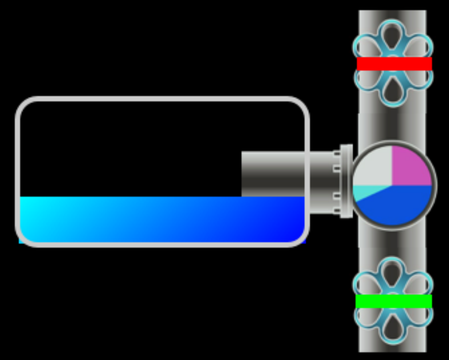


(A+B) index probe

Stages of Quick-Sorting

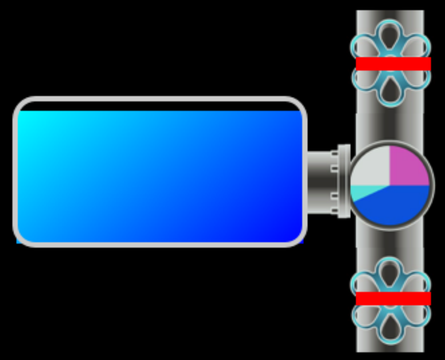
blocking

(1) reading



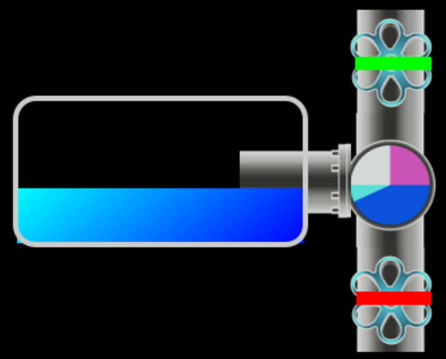
blocking

(2) sorting



non-blocking

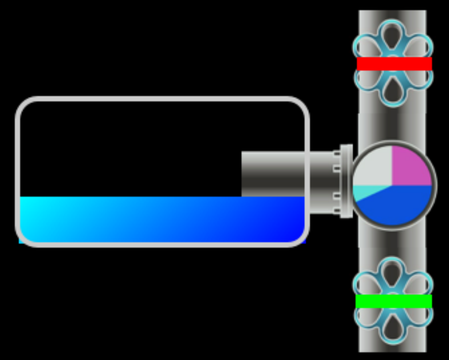
(3) outputting



Stages of External Merge Sort

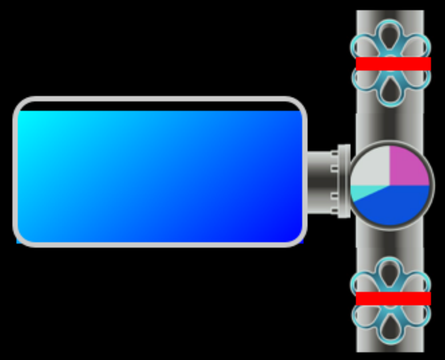
blocking

(1) reading
(with run generation)



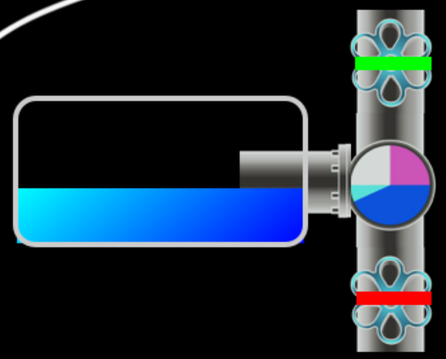
blocking

(2) merging
(without final merge)



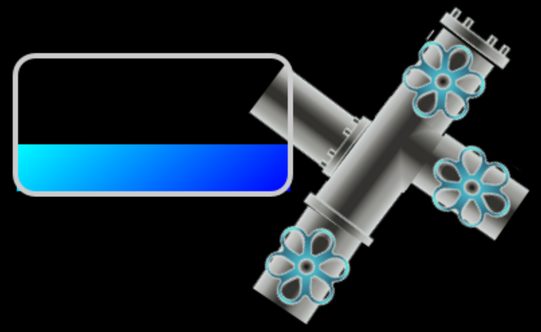
non-blocking

(3) outputting
(with final merge)

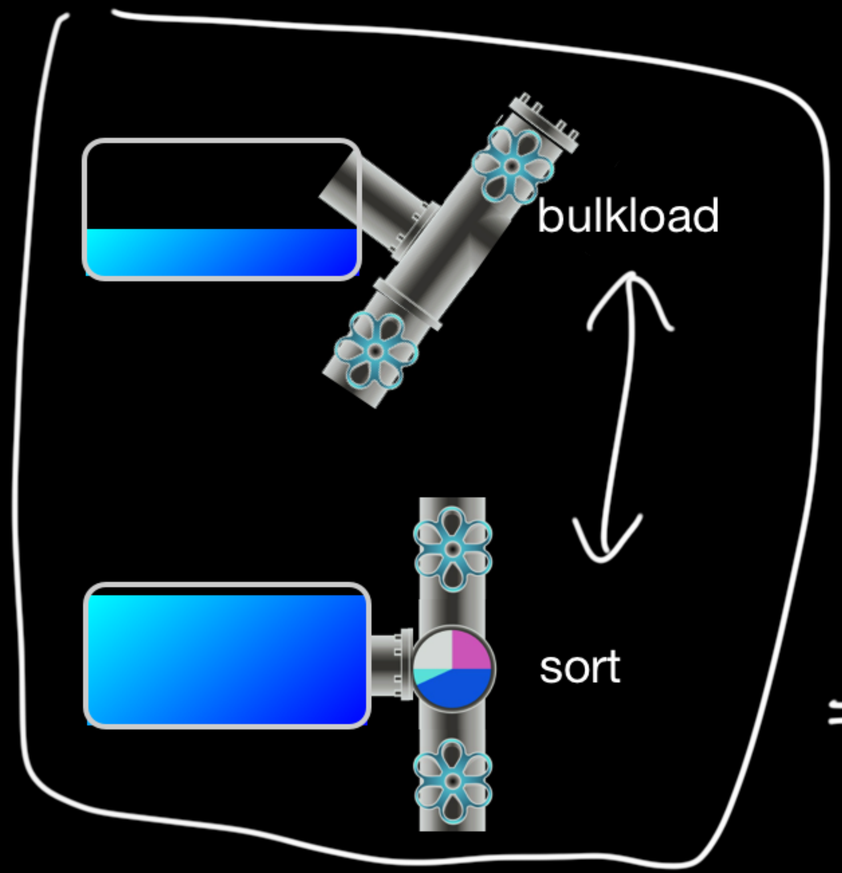


online-merge

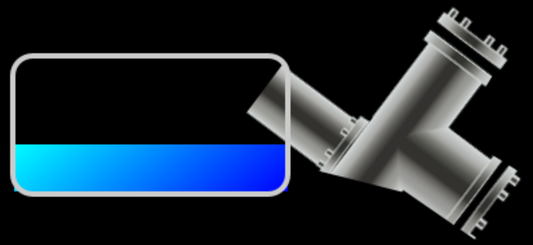
Blocking vs Non-Blocking Algorithms



INLJ/SHJ
(with bulkload)



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



INLJ/SHJ
(without bulkload)
- Index Probe

0 v 1

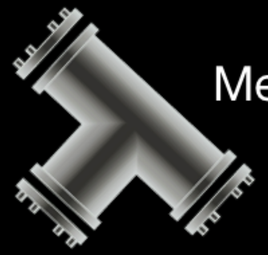
filter

1

1

project

1



Merge

here:
F=2
F≥2

MJ

DPHJ

← index probe

Copyrights and Credits