3000 drug. each with a note.

Aim: make drug pairs to find out it's relationship to a particular disease.

Original Algorithm:

Mapper: for drug 1

{drug 1, drug 2} – Note 1 → {drug 1, drug 2} – {Note 1, Note 2}

⋮

{drug 1, drug 3000} – Note 1 → {drug 1, drug 3000} – {Note 1, Note 3000}

{drug 2, drug 3} – {Note 2, Note 3}

⋮

Mapper: for drug 3000:

{drug 1, drug 3000} – Note 3000 →

⋮

{drug 2999, drug 3000} – Note 3000 → {drug 2999, drug 3000} – {Note 2999, Note 3000}

replicates !!

Cost: 3000 × 2999 × Size of Note. Too Much !!

Refine Algorithm:

Make groups: 30 group with each containing 100 drug.

Mappers work on the group unit.

G(i): number of group to which the drug i belongs.

key pair: {m, n}, m, n, number of groups

Mapper for drug i only produce 29 key-value pairs. like:

drug 1 in group 1, then have {1, 2} – Note 1

{1, 3} – Note 1

⋮

{1, 30} – Note 1.

drug 2 in group 1, then have {1, 2} – Note 2

⋮

{1, 30} – Note 2

In reducer, {1, 2} – {Note of all drug in group 1 & Note of all drug in group 2}

⇒ {1, 2} – list of Note of size 200.

Then the task is to compare each records in group 1 & each records in group 2. (total 200)

New cost 3000 × 29 × size of Note.