Dr. Bhargava's Part (DIA) work flow write up:

Problem: Optimization of SKU storage locations.

Objective: Implementing the combinatorial optimization problem on FPGA by creating a compressed graphical structure of ZDD (Zero Suppressed binary decision diagram)

Input: Cost of SKUs stored at Specific Locations.

SKU	Location	Cost
S1	L1	1234
S1	L2	2341
S1	L3	6753
--	--	--
S100	L99	1908
S100	L100	1876

Location Storage Capacity:

Location ID	Capacity in volume
L1	2345
L2	8436
--	--
L100	4535

SKU Volume

SKU	Volume
S1	1209
S2	8076
--	--
S100	5342

Note: SKU Volume and Capacity of Storage at location ID will be provided in same Units.
Output: Optimal location Identification for M SKUs based on previous days data.

Description: Based on specific time window, we would have the data for Order Put Information and Order Pick Information.
Order Put Information – This is an Inbound data. Items received to store and operate the warehouse.

Order Pick Information – This is an Order fulfilment data. How, when and from where the items are picked up from the warehouse to fulfil the orders.
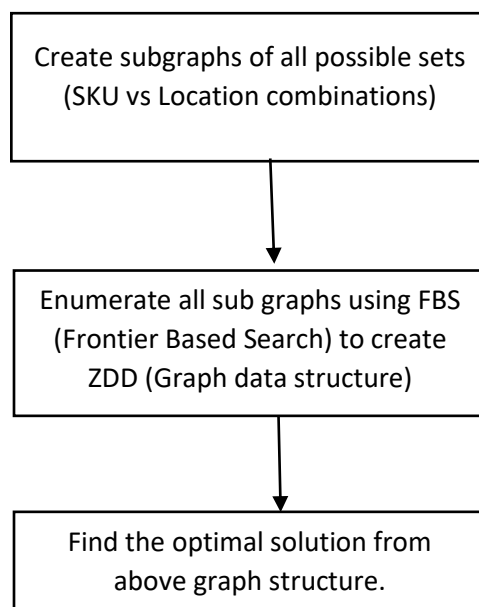
Mater Location data – This contains all locations and storage capacity of each location in the warehouse.

Master SKU data – Dimensions of all the SKUs.

Cost Calculation – The **Cost** above mentioned in the table is calculated based on all the above parameters. Pick, Put, relocation of certain items (In case if required), Distance travelled to pick from the depot.

Based on Cost, we decide to choose the best locations for the given SKUs.

Flow:

```
┌─────────────────────────────────┐
│  Create subgraphs of all         │
│  possible sets                   │
│  (SKU vs Location combinations)  │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│  Enumerate all sub graphs        │
│  using FBS                       │
│  (Frontier Based Search) to      │
│  create                          │
│  ZDD (Graph data structure)      │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│  Find the optimal solution from  │
│  above graph structure.          │
└─────────────────────────────────┘
```
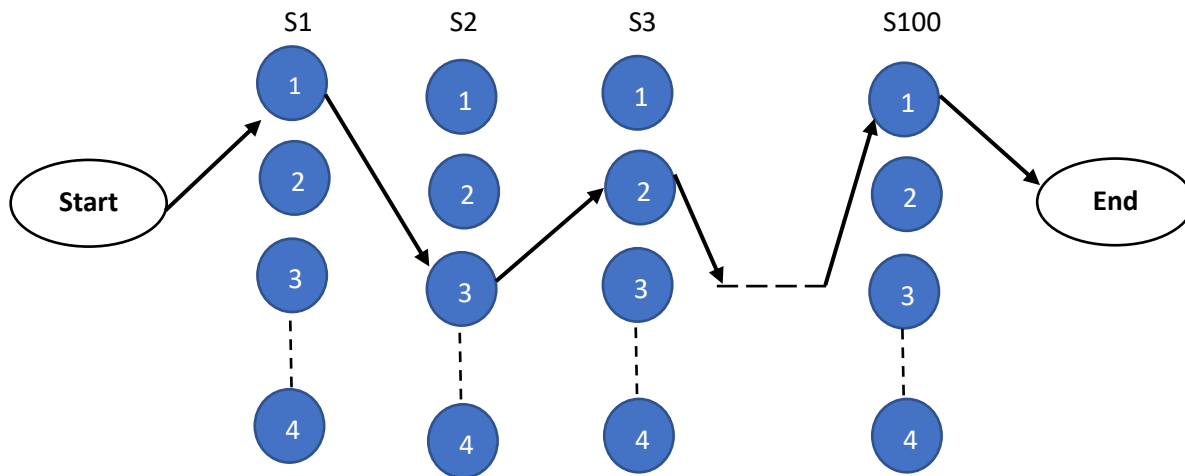
Expected Example flow of Working (Illustration purpose)

Start, End Indicate the start and end of the flow. (One possible combination of SKUs Locations 'start' and 'end'; for Illustration purpose only)

Blue color highlighted circles are locations

Numbers at the top for each vertical represent SKU ids. Ex: S1, S2, S3, … S100

Above flow indicates bunch of possible linear combinations. (At any given location we can place 'n' number of SKUs if the storage capacity at the given location is not full).

Expectation Result:

An implementation of above flow in FPGA to find the optimal solution.

1. A table for a given set of SKUs, respective locations.

SKU	Location
S1	L1
S2	L2
S3	L3
---	---
S100	L100

2. **Overall Cost** for the best combination observed.

Cost = Cost(S1, L1) + Cost(S2, L2) + Cost(S3, L3) + ----- + Cost(S100, L100)

Cost(S1,L1) --- This is the Cost to Store SKU S1 at Location L1 – This is provided as an Input.