

Code Documentation

This file contains explanation of code along with $O(n)$

Assumptions:

- I assume that only one product can be placed in one cell.
- I show all products inside product selection tab as customer can pick products from multiple stores.

Functions computational complexity:

- **getGrids()**
This function displays grid table on the screen.
 $O(n) = \text{Grid Height} * \text{Grid Width}$
- **placeProducts()**
This function place products on the grid table and inside product selection container.
 $O(n) = \text{Number of products} * \text{Number of stores}$
- **getStoreName()**
This function gets store name from database by store id
 $O(n) = \text{Number of stores}$
- **getSelectedProducts()**
This function calculates cell path in between selected products along with shortest route.
 $O(n) = \text{Number of selected products} + \text{findProductsPath}() O(n) + \text{shortestPathDynamic}() O(n)$
- **findProductsPath()**
This function calculates cell path between selected products.
 $O(n) = \text{Number of selected products} * \text{Number of grid cells} * \text{addRobotRoutes}()$
- **addRobotRoutes()**
This function calculates cell path route between selected products.
 $O(n) = \text{Number of grid cells} * 4$
- **shortestPathDynamic()**
This function calculates the shortest path from start point pick all products and get back to the starting point.
 $O(n) = \text{Number of selected products} * 2^{\text{Number of selected products}}$