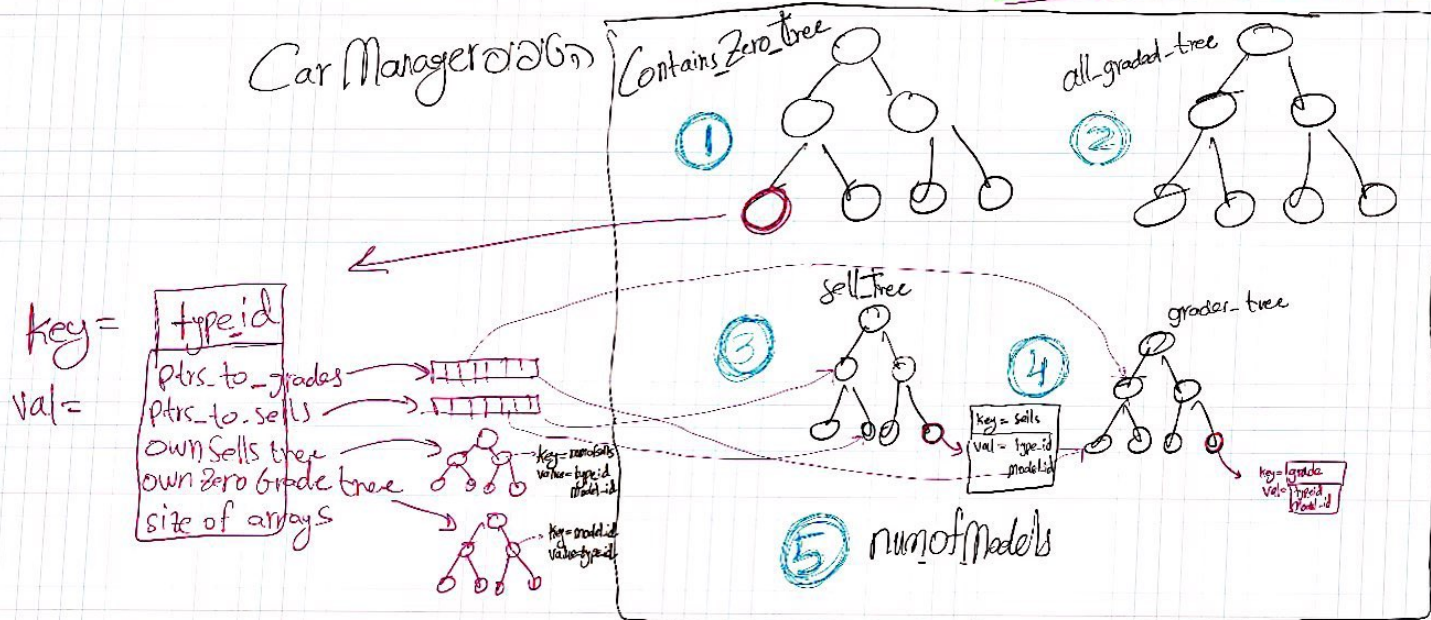


(1) NOT NO SORT OR AA



* כאשר המספר והמחיר של המכונית:

① Init():

1. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית

contains-zero-tree 11
all-graded-tree 12
sell-tree 13
grades-tree 14

2. $O(1)$: num of models שווה ל-0, המספר והמחיר של המכונית

② Add CarType (void* DS, int typeID, int numOfModels):

1. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
2. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
3. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
4. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית

NodeInfo

ptrs-to-grades-tree
ptrs-to-sells-tree
own-zero-grades-tree
own-sells-tree = null
size-of-array = numModels

key = 012... model-id
val = 666... type-id

$O(n)$: $carManager$ יוצר את המספר והמחיר של המכונית
 $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
 $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית

1. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
2. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
3. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית
4. $O(1)$: $carManager$ יוצר את המספר והמחיר של המכונית

SellCar(void** DS, int typeID, int modelID)

4

1. $AVL_{tree}(O(\log n))$ FAILURE: min & max of key = typeId of root of all good trees containing zero trees of size 0 or 1
 2. $AVL_{tree}(O(\log n))$ FAILURE: min & max of key = typeId of root of all good trees containing zero trees of size 0 or 1
 3. $AVL_{tree}(O(\log n))$ FAILURE: min & max of key = typeId of root of all good trees containing zero trees of size 0 or 1

[illegible]

MakeComplaint (void* DS, int typeID, int modeID, int t) (5)

• מילוט תחבולת הפיקוח של ההקצבה seller, בין השאר א"י, עומד מדינת ישראל (32,33), והעסקים של ה grade מהם מורידים $\left(-\frac{100}{T}\right)$ אף $(+10)$.

Selfish merge sort algorithm $O(\log n + \log M)$ \square

⑥

סוגי (types)
הנבדלות (differences) בין סוגי עצים (types of trees)
האיכות (quality) והכמות (quantity) של הענפים (branches)

Get Worst Models

7

1. במקרה של $O(1)$ אם $num_of_models < 0$ אז $O(1)$ FAILURE
 2. אם $num_of_models < 0$ אז $O(1)$ FAILURE

key = gene
val = type
model

grades-tree

3. bool * transfer = false, i = 0
 4. inorder ממוינת ממוינת (המיון)

2.1. אם $grade < 0$ אז $i < num_of_models$
 * (types+i) = type *
 * (models+i) = model *
 i++ *

2.2. אם $grade > 0$ אז $grade = 0$ אז $i < num_of_models$
 * transfer = false א/כ *
 * transfer = true *

3. grades-tree inorder ממוינת ממוינת (המיון)
 4. grades-tree inorder ממוינת ממוינת (המיון)

5. grades-tree inorder ממוינת ממוינת (המיון)
 6. grades-tree inorder ממוינת ממוינת (המיון)

7. grades-tree inorder ממוינת ממוינת (המיון)
 8. grades-tree inorder ממוינת ממוינת (המיון)

9. grades-tree inorder ממוינת ממוינת (המיון)
 10. grades-tree inorder ממוינת ממוינת (המיון)

