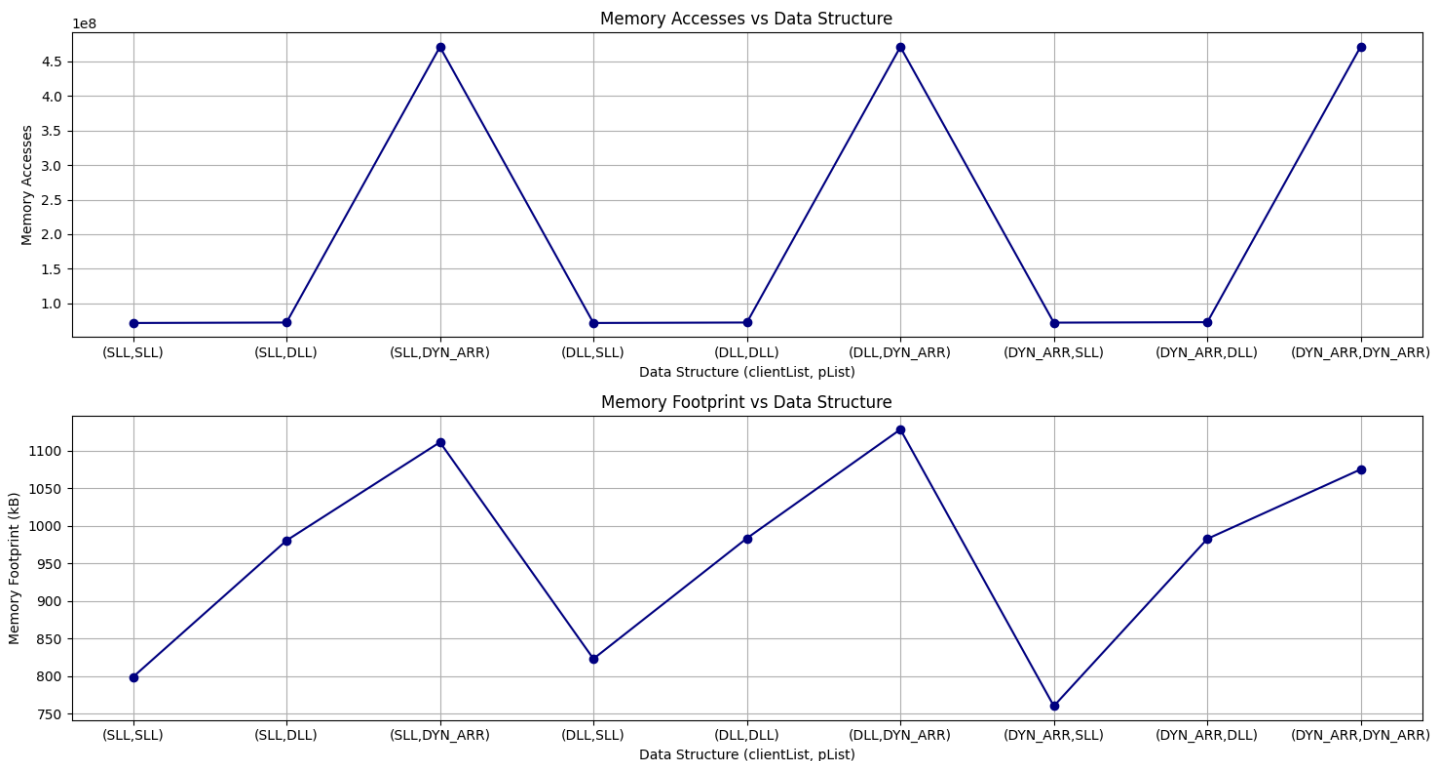


### Άσκηση 1: Βελτιστοποίηση δυναμικών δομών δεδομένων του αλγορίθμου DRR

Α) Εκτελούμε την εφαρμογή DRR με όλους τους διαφορετικούς τύπους δεδομένων (SLL, DLL, Dyn\_Arr) για τη λίστα των πακέτων και τη λίστα των κόμβων. Τα αποτελέσματα τα οποία λάβαμε για τα memory accesses και το memory footprint κάθε περίπτωσης, εμφανίζονται στα παρακάτω διαγράμματα, τα οποία δημιουργήθηκαν μέσω του `embedded 2.ipynb`:



Β) Παρατηρούμε από το παραπάνω διάγραμμα και από τις τιμές που περιέχονται στο αρχείο `embedded 2.ipynb`, ότι έχουμε τον μικρότερο αριθμό προσβάσεων στη μνήμη όταν τόσο η λίστα πακέτων όσο και η λίστα κόμβων υλοποιούνται ως *Single-Linked Lists*. Σε αυτή την περίπτωση, έχουμε **71817154 memory accesses**.

Γ) Παρατηρούμε, επίσης, ότι έχουμε τις μικρότερες απαιτήσεις σε μνήμη όταν η λίστα των κόμβων είναι υλοποιημένη ως *Dynamic Array* και η λίστα των πακέτων ως *Single-Linked List*. Σε αυτή την περίπτωση παρουσιάζεται **memory footprint** ίσο με **760.2 kB**.

## 2η Άσκηση – Σχεδιασμός Εναλλακτικών Συστημάτων

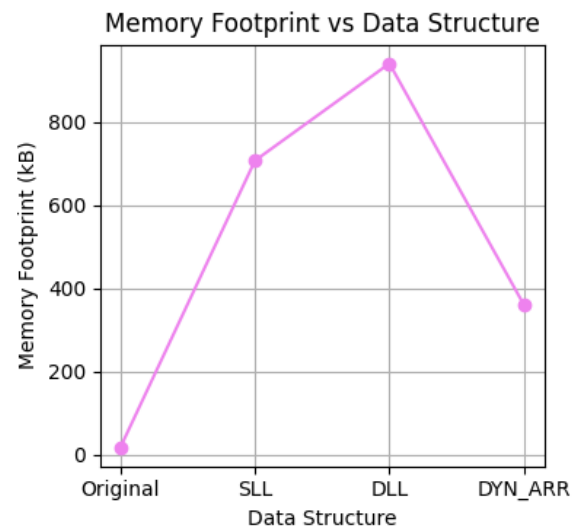
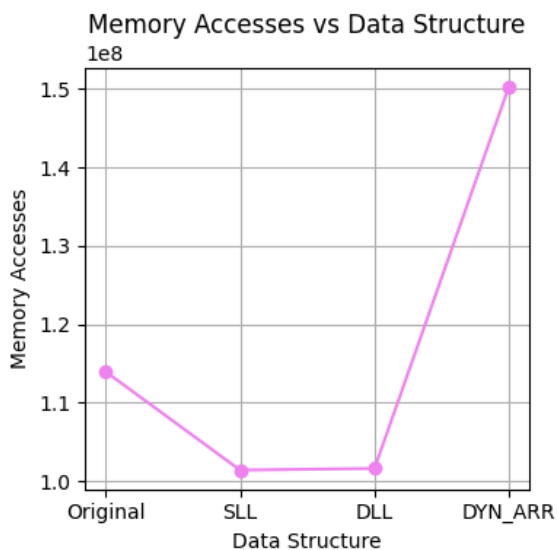
### Άσκηση 2: Βελτιστοποίηση δυναμικών δομών δεδομένων του αλγορίθμου Dijkstra

A) Εκτελούμε την εφαρμογή Dijkstra και τα αποτελέσματα που προκύπτουν είναι τα εξής:

```
Shortest path is 1 in cost. Path is: 0 41 45 51 50
Shortest path is 0 in cost. Path is: 1 58 57 20 40 17 65 73 36 46 10 38 41 45 51
Shortest path is 1 in cost. Path is: 2 71 47 79 23 77 1 58 57 20 40 17 52
Shortest path is 2 in cost. Path is: 3 53
Shortest path is 1 in cost. Path is: 4 85 83 58 33 13 19 79 23 77 1 54
Shortest path is 3 in cost. Path is: 5 26 23 77 1 58 99 3 21 70 55
Shortest path is 3 in cost. Path is: 6 42 80 77 1 58 99 3 21 70 55 56
Shortest path is 0 in cost. Path is: 7 17 65 73 36 46 10 58 57
Shortest path is 0 in cost. Path is: 8 37 63 72 46 10 58
Shortest path is 1 in cost. Path is: 9 33 13 19 79 23 77 1 59
Shortest path is 0 in cost. Path is: 10 60
Shortest path is 5 in cost. Path is: 11 22 20 40 17 65 73 36 46 10 29 61
Shortest path is 0 in cost. Path is: 12 37 63 72 46 10 58 99 3 21 70 62
Shortest path is 0 in cost. Path is: 13 19 79 23 77 1 58 99 3 21 70 55 12 37 63
Shortest path is 1 in cost. Path is: 14 38 41 45 51 68 2 71 47 79 23 77 1 58 33 13 92 64
Shortest path is 1 in cost. Path is: 15 13 92 94 11 22 20 40 17 65
Shortest path is 3 in cost. Path is: 16 41 45 51 68 2 71 47 79 23 77 1 58 33 32 66
Shortest path is 0 in cost. Path is: 17 65 73 36 46 10 58 33 13 19 79 23 91 67
Shortest path is 1 in cost. Path is: 18 15 41 45 51 68
Shortest path is 2 in cost. Path is: 19 69
```

B) Ο τροποποιημένος κώδικας του `dijkstra.c` που επισυνάπτεται, αντικαθιστά τη δομή δεδομένων με τις δομές δεδομένων της βιβλιοθήκης. Για κάθε δομή τα αποτελέσματα αποθηκεύονται στο αντίστοιχο αρχείο XXX.txt. Στη συνέχεια, μέσω του `embedded_2.ipynb`, το περιεχόμενο των αρχείων συγκρίνεται με τα αποτελέσματα του ερωτήματος (A) (`original.txt`) και διαπιστώνουμε ότι είναι ίδια.

Γ) Εκτελούμε την εφαρμογή με όλους τους διαφορετικούς τύπους δεδομένων (SLL, DLL, Dyn\_Arr). Τα αποτελέσματα τα οποία λάβαμε για τα `memory accesses` και το `memory footprint`, εμφανίζονται στα παρακάτω διαγράμματα:



## 2η Άσκηση – Σχεδιασμός Ενωματωμένων Συστημάτων

Δ) Παρατηρούμε ότι η εφαρμογή έχει τον μικρότερο αριθμό προσβάσεων στη μνήμη (**101437372**), όταν η δομή δεδομένων είναι *Single-Linked List*.

Ε) Παρατηρούμε ότι η εφαρμογή έχει τις μικρότερες απαιτήσεις μνήμης (**360.6 KB**), όταν η δομή δεδομένων είναι *Dynamic Array* (εξαιρώντας την αρχική υλοποίηση της εφαρμογής).