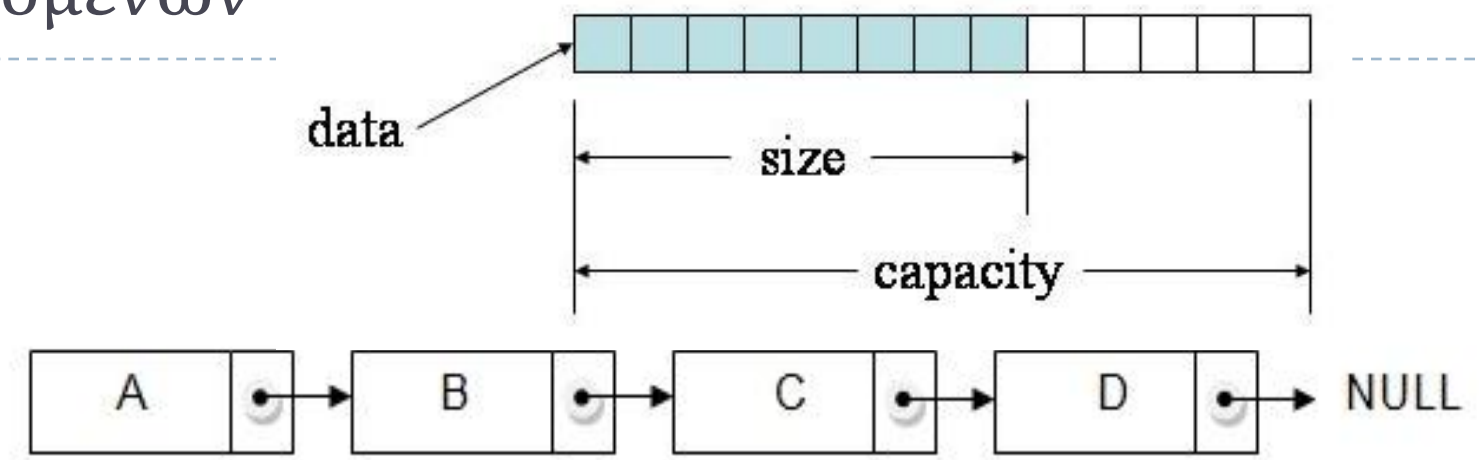
Θεμελιώδη Θέματα Επιστήμης Υπολογιστών

Ασκήσεις στη Βελτιστοποίηση Δυναμικών Δομών Δεδομένων -

Μεθοδολογία Βελτιστοποίησης Δυναμικών Δομών Δεδομένων



Single Linked List

Δομή Δεδομένων (Data Structure)	Προσβάσεις στη μνήμη για τυχαίο στοιχείο (memory accesses for random element)	Μέγεθος μνήμης (Memory Footprint)
Απλά Συνδεδεμένη Λίστα (Single Linked List)	Πολλές (O(n))	Μικρό (παρόλο που απαιτούνται +32bit για κάθε element)
Δυναμικός Πίνακας (Dynamic Array)	Μία (O(1))	Μεγάλο (διπλασιάζει το μέγεθός του κάθε φορά που γεμίζει)

Μεθοδολογία Βελτιστοποίησης Δυναμικών Δομών Δεδομένων

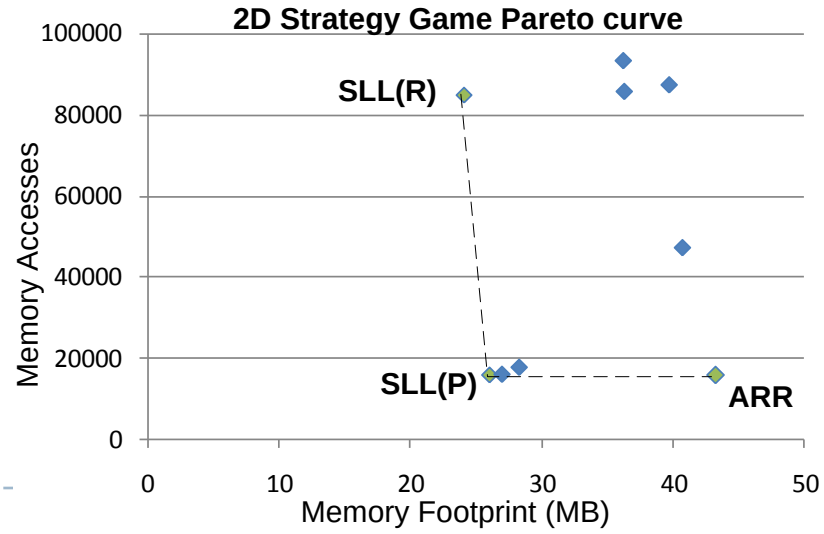
- Το είδος της δομής δεδομένων που θα επιλέξουμε επηρεάζει:
  - Τον **αριθμό των προσβάσεων στη μνήμη** (memory accesses) της **εφαρμογής**.
  - Το **μέγεθος της μνήμης** (memory footprint) που απαιτεί η **εφαρμογή**.
  - Άρα, κατά συνέπεια και την **απόδοση** (performance) και την **κατανάλωση ενέργειας** (energy consumption) ολόκληρου του συστήματος.

Μεθοδολογία Βελτιστοποίησης Δυναμικών Δομών Δεδομένων

- Εφαρμογή (Application)

- Ενσωματωμένο σύστημα (embedded system)

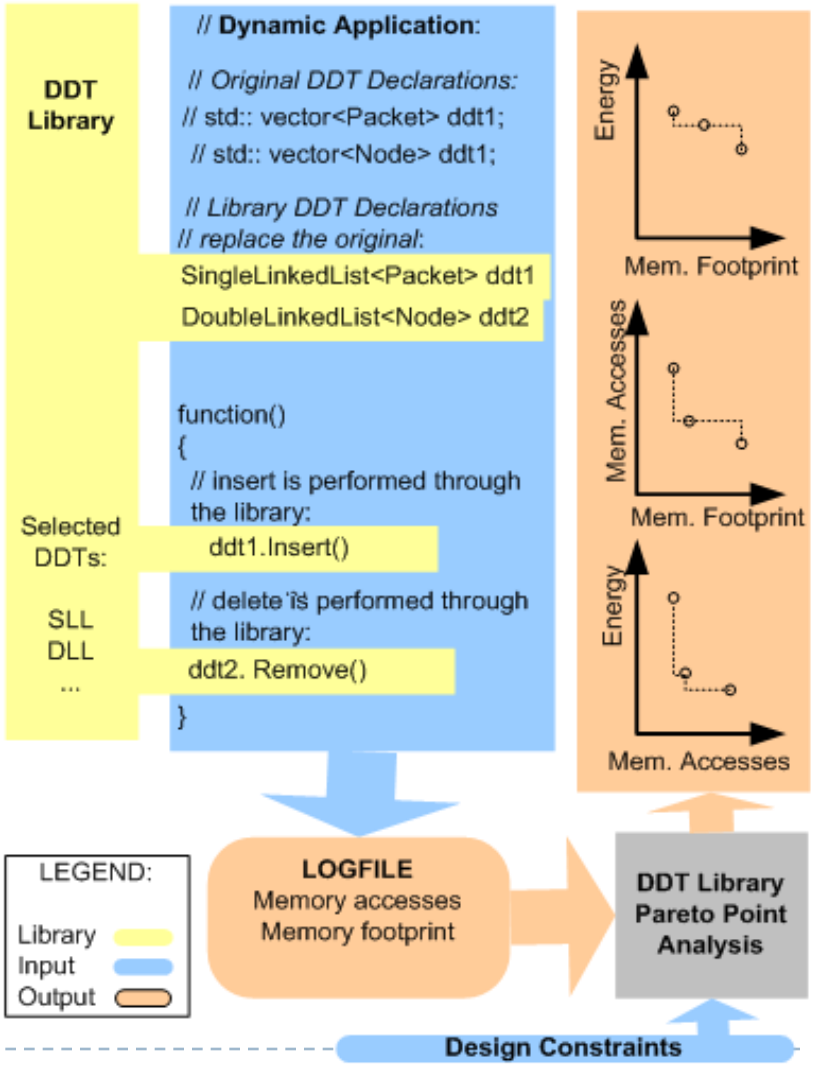
- Constraint: fps: 30fps



- Constraint: Memory size: 1GB

Μεθοδολογία Βελτιστοποίησης Δυναμικών Δομών Δεδομένων

- Βιβλιοθήκη δυναμικών δομών δεδομένων:
  - **Αντικαθιστούμε τις δομές δεδομένων της εφαρμογής με αυτές της βιβλιοθήκης.**
  - **Τρέχουμε την εφαρμογή κάθε φορά για διαφορετική δομή δεδομένων που επιλέγουμε από τη βιβλιοθήκη και μετράμε προσβάσεις και μέγιστο μέγεθος μνήμης για κάθε μία.**



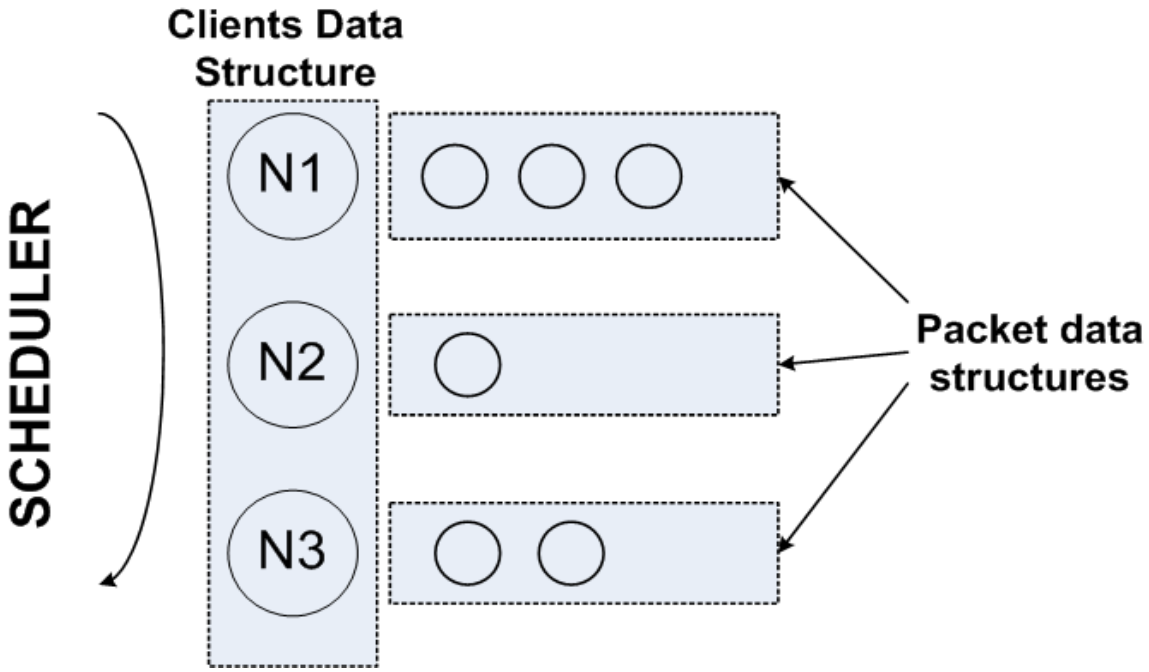
Directories

- Η βιβλιοθήκη δυναμικών δομών δεδομένων: synch_implementations.
- Εφαρμογή DRR
- Εφαρμογή Dijkstra

DRR: Deficit Round Robin

- Αλγόριθμος δρομολόγησης πακέτων.
- Κάθε κόμβος έχει ένα πεδίο “deficit”
- Κάθε φορά που ο scheduler επισκέπτεται έναν κόμβο αυξάνει το deficit κατά Quantum.
  - Αν το μέγεθος του πρώτου πακέτου προς προώθηση είναι μεγαλύτερο του deficit, τότε το πακέτο προωθείται και το deficit του κόμβου μειώνεται κατά το packet size.
  - Αν όχι, ο scheduler προχωράει στον επόμενο κόμβο.

DRR: Deficit Round Robin



- Δομές δεδομένων:
  - Η δομή δεδομένων των κόμβων
  - Η δομή δεδομένων των πακέτων κάθε κόμβου.

Διαδικασία εισαγωγής της βιβλιοθήκης στον κώδικα της εφαρμογής

1. Include Library files

   #include "../synch_implementations/cdsl_sll.h

2. Declaration of the data structures

   cdsl_sll *clientList;

3. Initialization

   clientList = cdsl_sll_init();

4. Replacement of data structure operations

   clientList→enqueue(0, clientList, (void*)mydata);

   data = clientList → dequeue(0, clientList);

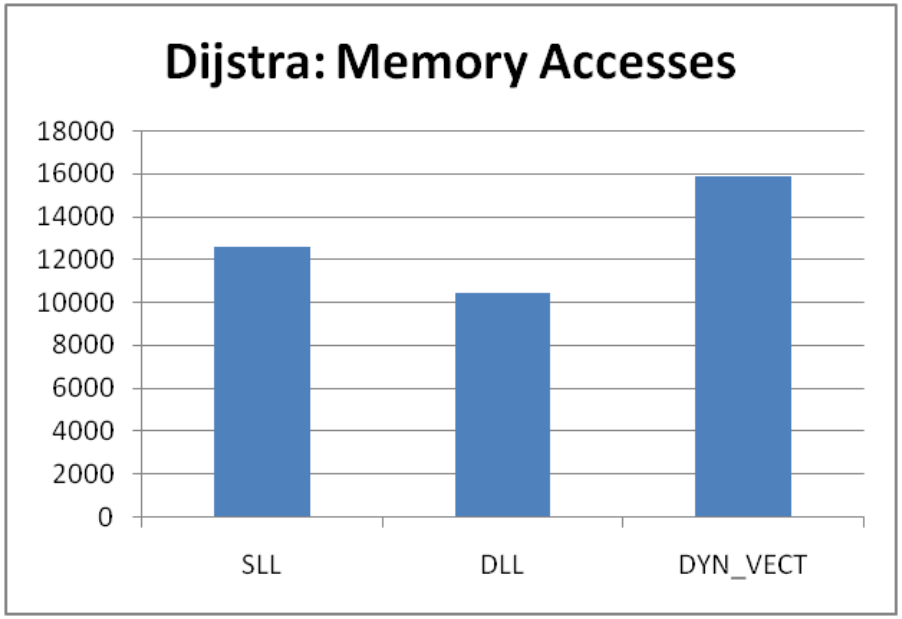
Dijkstra

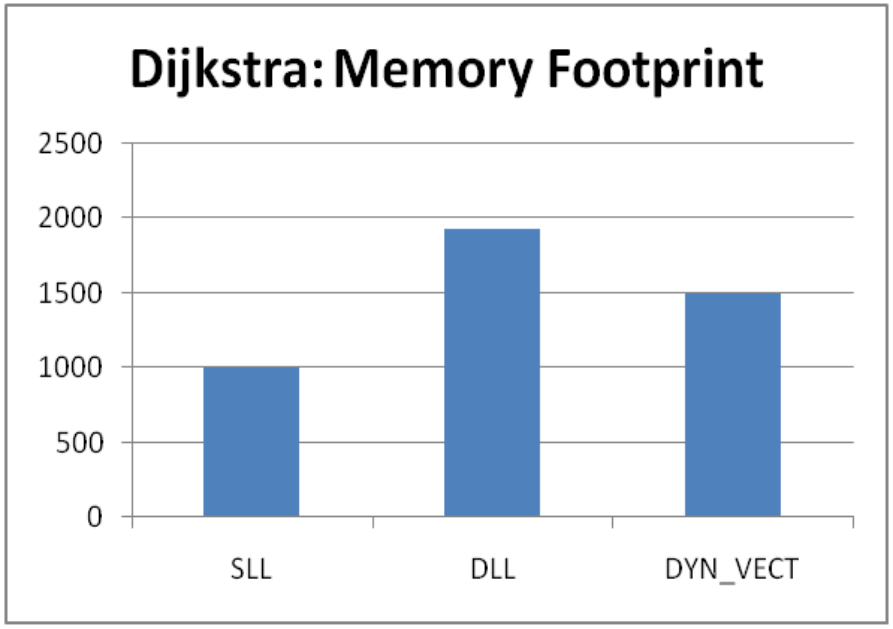
- Δικτυακός αλγόριθμος εύρεσης διαδρομής με μικρότερο κόστος.
- Δομή δεδομένων:
  - Μία queue στην οποία αποθηκεύονται οι κόμβοι που εξετάζει.
- Το βήμα (4) θα γίνει στις συναρτήσεις:
  - enqueue(): Εισάγει έναν νέο κομβο στο τέλος της λίστας
  - dequeue(): Αφαιρεί τον πρώτο κόμβο από τη λίστα

Εξαγωγή αποτελεσμάτων

- Εγκαθιστάτε τη Valgrind Suite.
- Θα χρησιμοποιήσετε τα εργαλεία Massif(memory footprint) και Lackey (memory accesses).

Παρουσίαση αποτελεσμάτων





- **…και το ίδιο για το DRR.**
- **Προσοχή: 9 συνδυασμοί: 3 για τη δομή δεδομένων των clients x 3 για τη δ.δ. των πακέτων.**