Amortization w/ potential method

10/30/24

Recall: Amortized analysis

- Worst case over a sequence of operations

- Method 1: Aggregate method
  - Compute time for n operations and divide by n

- Method 2: Accounting method
  - Run data structures “store”; amortized cost is what you charge, pay for work
  - Put extra money “into” the data structure and show there is always enough

- Method 3: Potential method
  - Put extra money into one account, the potential Φ
  - $\Phi = 0$ initially and $\Phi >= 0$ always
  - amortized cost = actual cost + $\Delta\Phi$ = actual cost + $\Phi_{after} - \Phi_{before}$

Unordered linked lists

- Consider set operations on an unordered linked list
  - access/find: search list for desired value
  - insert: find first to make sure the value isn't there already
  - delete: find and remove the value

- Cost model:
  - accessing i^{th} value costs i
  - can move a value toward the front for free in an access
  - swapping any other adjacent pair costs 1

Move-to-front (MTF)

Heuristic: When you access an element, move it to the front of the list

- Insert values in the front if they're not in the list (or move them if they are)

Theorem: For any sequence of operations, MTF has at most twice the cost of any other algorithm

Looking at a sequence of operations

$c_{OPT}(t)$ = cost to optimal algorithm for t^{th} access

$c_{MTF}(t)$ = cost to MTF for t^{th} access

Looking at a sequence of operations

$c_{OPT}(t)$ = cost to optimal algorithm for t^{th} access

$c_{MTF}(t)$ = cost to MTF for t^{th} access

Φ = number of inversions between MTF's list and optimal list

$\Phi(t)$ = number of inversions after the t^{th} access

Meets the conditions:

$\Phi(0) = 0$

$\Phi(t) \geq 0$

Claim: $C_{MTF}(t) + \Phi(t) - \Phi(t-1) \leq 2C_{OPT}(t) - 1$

Suffices to prove the claim since summing it over m moves gives

$$\text{MTF cost} + \Phi(m) \leq 2 \text{ OPT cost} - m$$

Claim: $C_{MTF}(t) + \Phi(t) - \Phi(t-1) \leq 2C_{OPT}(t) - 1$

Proof:

Let x_t be the item accessed in t^{th} step.

Let k = #items before x_t in both lists

Let i = #items before x_t in MTF, but after it in OPT

Claim: $C_{MTF}(t) + \Phi(t) - \Phi(t\text{-}1) \leq 2C_{OPT}(t) - 1$

Proof:

Let x_t be the item accessed in t^{th} step.

Let k = #items before x_t in both lists

Let i = #items before x_t in MTF, but after it in OPT

What $C_{MTF}(t)$, the cost of MTF accessing x_t and moving it to the front?

A. k
B. k+1
C. i
D. k+i
E. k+i+1

Cost model:

accessing i^{th} value costs i (free to move it toward front)
swapping any other adjacent pair costs 1

Claim: $C_{MTF}(t) + \Phi(t) - \Phi(t-1) \leq 2C_{OPT}(t) - 1$

Proof:

Let x_t be the item accessed in t^{th} step.

Let k = #items before x_t in both lists

Let i = #items before x_t in MTF, but after it in OPT

What $C_{MTF}(t)$, the cost of MTF accessing x_t and moving it to the front?

A. k
B. k+1
C. i
D. k+i
E. k+i+1

Cost model:

accessing i^{th} value costs i (free to move it toward front)

swapping any other adjacent pair costs 1