71

Tuesday, October 22, 2019 2:11 PM

Potential Function Method

"cheap" operations store energy for "expensive" operations

Define a potential $\Phi(D_i)$ such that $\Phi(D_0) = 0$ and (*)

$\forall i \ \Phi(D_0) \geq 0$ (**)

Define amortized complexity σ_i as $a_i = t_i + \overbrace{\Phi(D_i) - \Phi(D_{i-1})}^{\Delta\Phi(D_i)}$

Proposition: Under (*) and (**):

$$\sum_{i=1}^{m} t_i \leq \sum_{i=1}^{m} a_i$$

Proof:

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} t_i + \Phi(D_i) - \Phi(D_{i-1})$$

$$= \sum_{i=1}^{m} t_i + \left(\Phi(D_m) - \Phi(D_0)\right)$$

$$= \sum_{i=1}^{m} t_i + \Phi(D_m)$$

$$\geq \sum_{i=1}^{m} t_i$$

∎

Example: Binary Counters

Let $A[0..k-1]$ be the binary counter

A | 1. . . . . | 0 | . . . | Define $\Phi(A) = \left|\{i : A[i] = 1\}\right|$

$A \; |\underbrace{1 \cdots\cdots}_{j}|0| \cdots |$ Define $\Phi(A) = |\{i : A[i] = 1\}|$

then $t_i = j+1$ bit flips. $\Delta\Phi = -j+1$, then

$$a_i = t_i + \Delta\Phi = j+1 + (-j+1) = 2$$

Example: Dynamic Tables

T is a table (hash table, binary heap).

$T.size$: # slots in the table

$T.num$: # items stored in T

$\alpha(T) = T.num \,/\, T.size$

$\frac{1}{4}$?

want $\frac{1}{2} \leq \alpha(T) \leq 1$ during Insert / Delete ops

Operations:

- Insert (T, x): If $\alpha(T) < 1$, then insert x into T
  If $\alpha(T) = 1$, allocate a new table of size $2(T.size)$, copy the elements, then insert x. Finally, release old space.

$t_i = 1$ (for new element inserted) +
1 for every element copied

$$\Phi(T_i) = 2(T.num - T.size/2) = 2(T.num) - T.size$$

$$a_i = t_i + \Phi(T_i) - \Phi(T_{i-1})$$

No expansion: $t_i = 1$ $\quad \Delta\Phi(T_i) = 2$ $\quad \Rightarrow \quad a_i = 3$

Expansion $\quad t_i + T_{i-1}.size$

$$\Phi(T_i) = 2 - T_{i-1}.size \quad \Rightarrow \quad a_i = 3$$