CS38
Introduction to Algorithms

Lecture 6

April 17, 2014

Outline

- data structures for MST and Dijkstra's
  - union-find with $\log^* n$ analysis (finishing up)
  - amortized analysis: potential function method
  - Fibonacci heaps

Recall: amortized analysis

- amortized analysis:
  - each operation has an amortized cost
  - any sequence of operations has cost bounded by sum of amortized costs

1. Fibonacci heap amortized vs. binary heap
   - EXTRACT-MIN $O(\log n)$ vs. $O(\log n)$
   - INSERT, DECREASE-KEY $O(1)$ vs. $O(\log n)$

Potential function method

- n operations on initial data structure D_0
  - D_i after i-th operation
- potential function $\Phi(D_i)$ (real number)
- amortized cost of i-th operation w.r.t Φ:

$$\underline{\mathbf{c_i}} = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

amortized cost

actual cost

change in potential

Potential function method

- n operations on initial data structure D_0
  - D_i after i-th operation
- potential function $\Phi(D_i)$ (real number)
- amortized cost of i-th operation w.r.t Φ:

$$\underline{\mathbf{c_i}} = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

- Key observation:

$$\sum_{i=1}^{n} \underline{\mathbf{c_i}} = \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$$

Potential function method

- Key observation:

$$\sum_{i=1}^{n} \underline{\mathbf{c_i}} = \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$$

  - sum of amortized costs is an upper bound on sum of actual costs, provided:

$$\Phi(D_n) - \Phi(D_0) \geq 0$$

  - will typically ensure $\Phi(D_i) - \Phi(D_0) \geq 0$ for all i

Potential function method

- Example: binary counter C on k bits
  - single operation: INCREMENT

    1010010001110111101**0**1111111 → 1010010001110111101**1**000000

  - worst case cost?

Potential function method

- Example: binary counter C on k bits
  - single operation: INCREMENT

    1010010001110111101**0**1111111 → 1010010001110111101**1**000000
- Potential function: $\Phi(C)$ = # of ones
- Consider i-th operation: t_i
  - actual cost c_i = (# of ones set to zero) +1
  - $\Delta\Phi = (\Phi(C_{i-1}) - t_i + 1) - \Phi(C_{i-1}) = 1 - t_i$
  - so amortized cost $\underline{c_i} = 2$

Potential function method

- Example: binary counter C
  - single operation: INCREMENT

    1010010001110111101**0**1111111 → 1010010001110111101**1**000000
- Starting with 0 counter on k bits:
  - $\Phi(C_0) = 0$, $\Phi(C_i) \geq 0$ for all i
  - so total cost of n operations is 2n
- Starting with arbitrary value on k bits:

  $\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \underline{c_i} - \Phi(C_n) + \Phi(C_0) \leq 2n + k$

Fibonacci heaps data structure

Kevin Wayne's slides
based on CLRS text