CSE 674 Advanced Data Structures

Vectors, Deques; Performance on Lists

Andrew C. Lee

EECS, Syracuse

Contents

- The *vector* data structure
- Double ended Queues: *deques*
- Amortized Analysis

C++ Vectors

- C++: Object Oriented; with STL (Standard Template Library)
- Some data structures are implemented as containers in STL: `array`, `deque`, `list`, `stack`, `queue`, `set`, etc ..
- vector is one of the containers
- Reference:
  1. Drozdek, Chapter 1, Section $7 - 8$
  2. `http://www.cplusplus.com/reference/stl/`

Vectors in C++ STL

- Stores in contiguous blocks of memory
- "Flexible" memory management:
  - See Example Handouts
  - Drozdek Figure 1.4

Reference: Drozdek Ch.1

Amortized Analysis

Question: How to assess the performance of vector operations ?

1. We want to obtain the *average performance* of each data structure operations in the worse case
2. Suppose $T(n)$is an upper bound for *any* sequence of n data structure operations We are interested in:

$$\frac{T(n)}{n} = \text{ The average cost per operation}$$

3. Example: Stack operations (Cormen, Chapter 17.1 - 17.2)

Deques

1. Another containers in C++ STL
2. It is a double-ended queue (generalized stacks and queues)
3. Examples from Handouts
4. Examples from Drozdek Figure 4.20 and Figure 4.21