

# Dynamic Memory Allocation II

## March 27, 2008

### Topics

- Explicit doubly-linked free lists
- Segregated free lists
- Garbage collection
- Review of pointers
- Memory-related perils and pitfalls

# Keeping Track of Free Blocks

**Method 1:** Implicit list using lengths -- links all blocks



**Method 2:** Explicit list among the free blocks using pointers within the free blocks



**Method 3:** Segregated free lists

- Different free lists for different size classes

**Method 4:** Blocks sorted by size (not discussed)

- Can use a balanced tree (e.g. Red-Black tree) with pointers within each free block, and the length used as a key

# Explicit Free Lists

Maintain list(s) of *free* blocks, not *all* blocks

- The “next” free block could be anywhere
  - So we need to store forward/back pointers, not just sizes
- Still need boundary tags for coalescing
- Luckily we track only free blocks, so we can use payload area



Note: links are generally not in the same order as the blocks!

