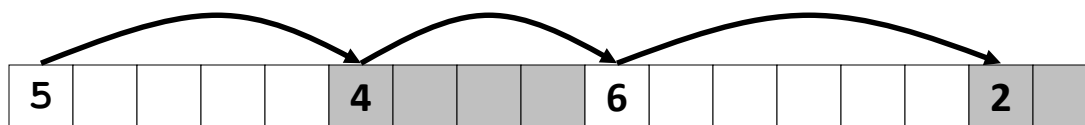


Keeping Track of Free Blocks

- Method 1: *Implicit list* using length—links all blocks



Assignment Project Exam Help

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



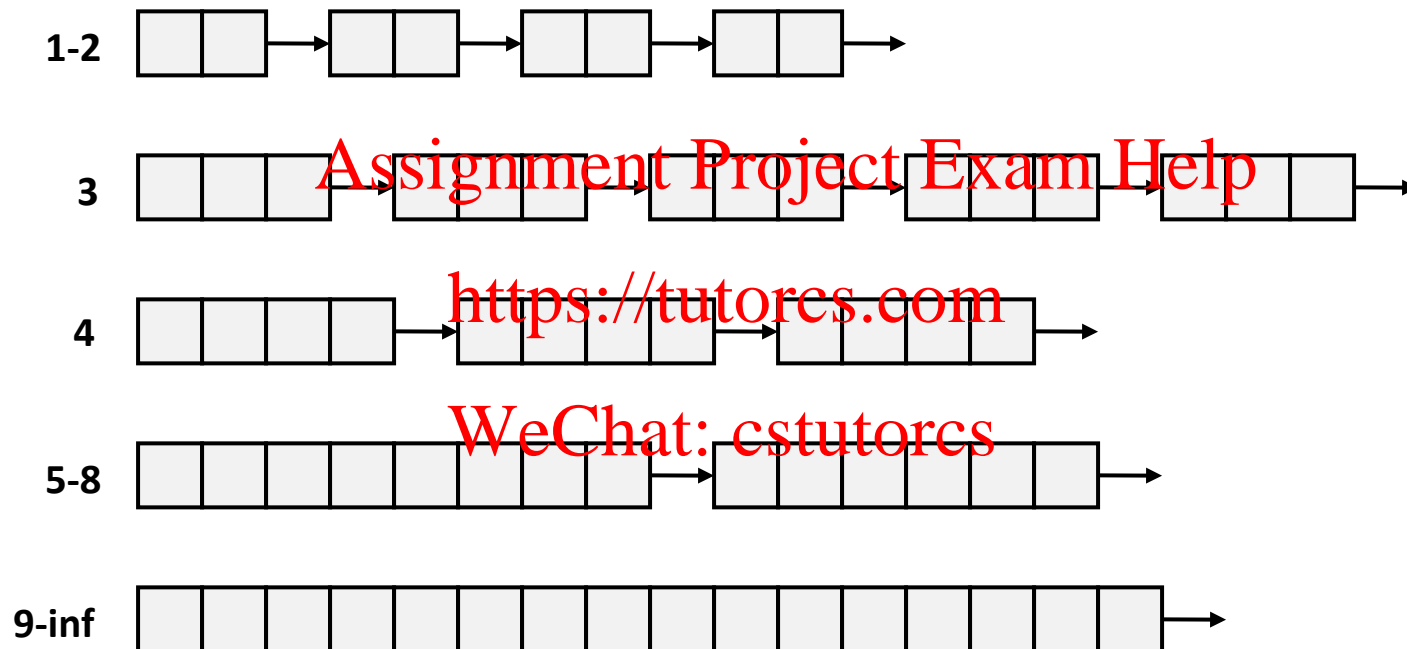
- Method 3: *Segregated free list*
 - Different free lists for different size classes

- Method 4: *Blocks sorted by size*

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

Segregated List (Seglist) Allocators

- Each *size class* of blocks has its own free list



- Often have separate classes for each small size
- For larger sizes: One class for each two-power size

Seglist Allocator

- Given an array of free lists, each one for some size class
- To allocate a block of size n :
 - Search appropriate free list for block of size $m \geq n$
 - If an appropriate block is found:
 - Split block and place fragment on appropriate list (optional)
 - If no block is found, try next larger class
 - Repeat until block is found
- If no block is found:
 - Request additional heap memory from OS (using `sbrk()`)
 - Allocate block of n bytes from this new memory
 - Place remainder as a single free block in largest size class.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Seglist Allocator (cont.)

■ To free a block:

- Coalesce and place on appropriate list (optional)

■ Advantages of seglist allocators

- Higher throughput
 - log time for power-of-two size classes
- Better memory utilization
 - First-fit search of segregated free list approximates a best-fit search of entire heap.
 - Extreme case: Giving each block its own size class is equivalent to best-fit.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs