Memory Allocation II

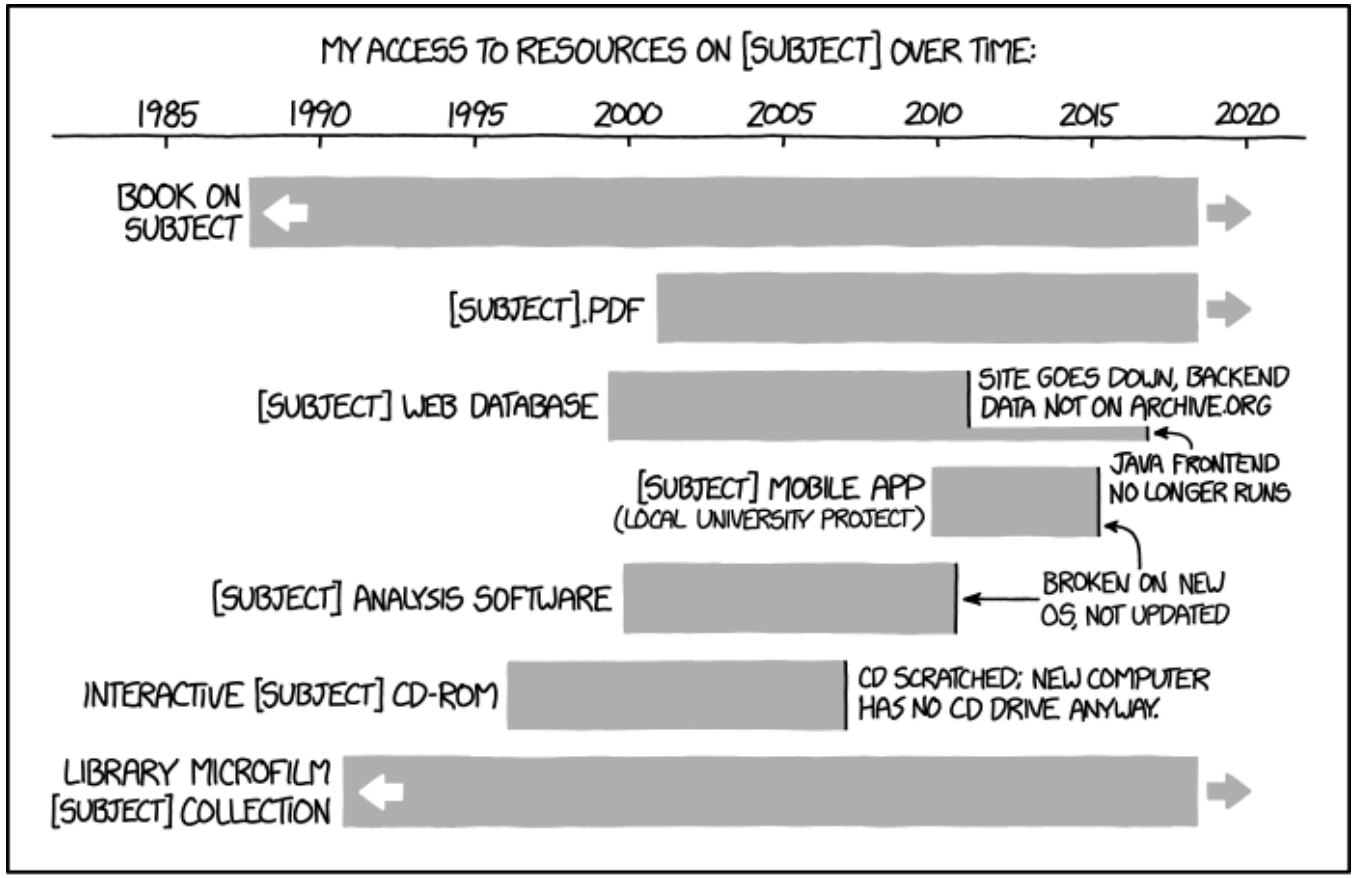
CSE 351 Autumn 2023

Instructor:

Justin Hsia

Teaching Assistants:

Afifah Kashif
Bhavik Soni
Cassandra Lam
Connie Chen
David Dai
Dawit Hailu
Ellis Haker
Eyoel Gebre
Joshua Tan
Malak Zaki
Naama Amiel
Nayha Auradkar
Nikolas McNamee
Pedro Amarante
Renee Ruan
Simran Bagaria
Will Robertson



IT'S UNSETTLING TO REALIZE HOW QUICKLY DIGITAL RESOURCES CAN DISAPPEAR WITHOUT ONGOING WORK TO MAINTAIN THEM.

http://xkcd.com/1909/

Relevant Course Information

- HW20 due Monday (11/20)
- HW21 due Friday (11/24)
  - Another double homework, but mostly about Lesson 21 (all but last slide)
  - Probably want to finish by 11/22

- Lab 4 due Monday after Thanksgiving (11/27)
- Lab 5 (Mem Alloc) will be released on Monday (11/20)

Memory Allocation II

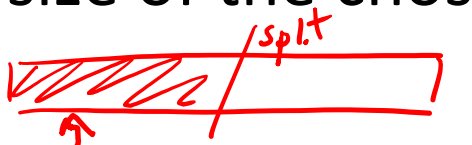
Fulfilling an Allocation Request

1) Compute the necessary block size payload + metadata + padding

2) Search for a suitable free block using the allocator's *allocation strategy*
   - If found, continue
   - If not found, return NULL

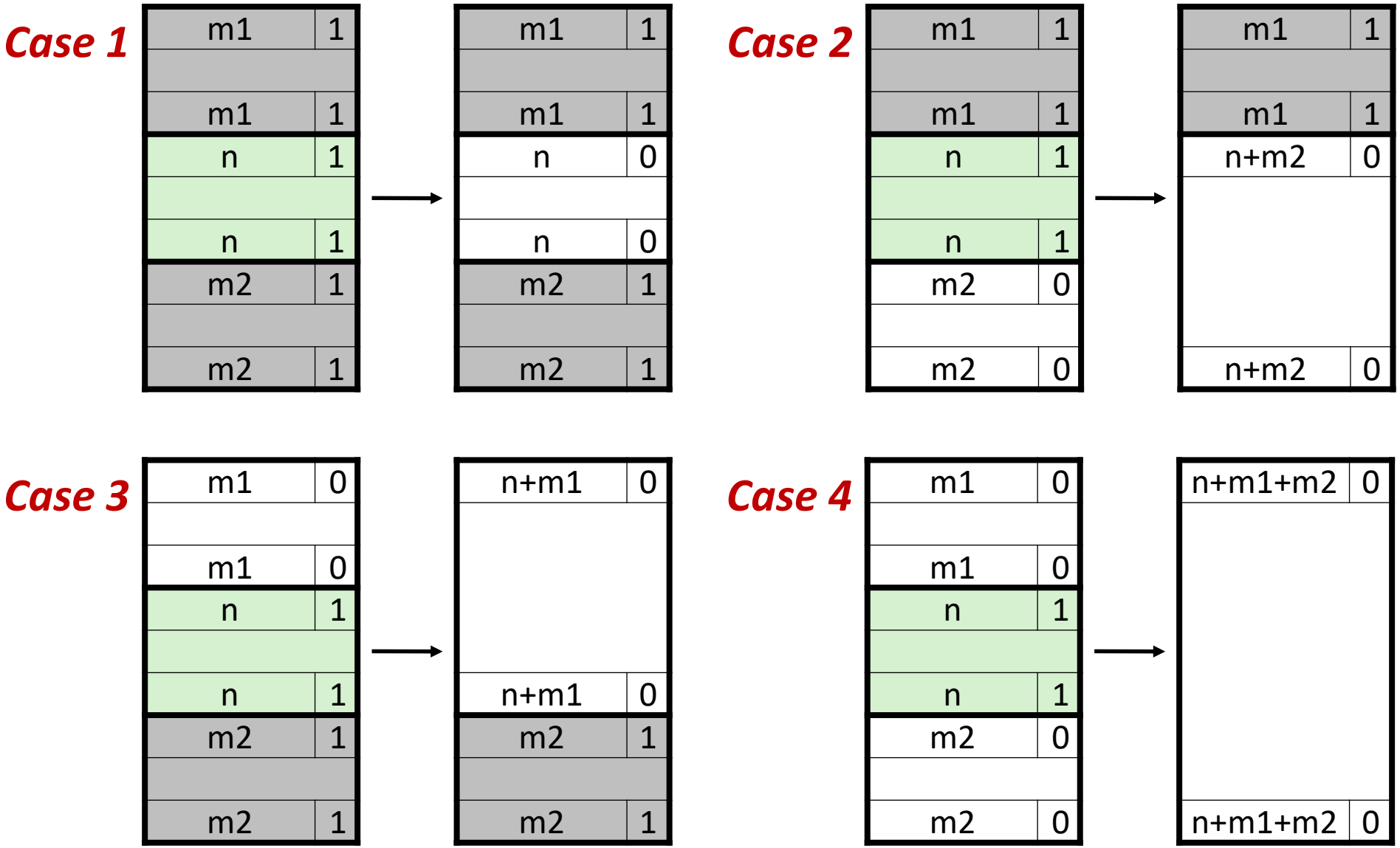
first fit
next fit
best fit

3) Compare the necessary block size against the size of the chosen block
   - If equal, allocate the block
   - If not, *split* off the excess into a new free block before allocating the block

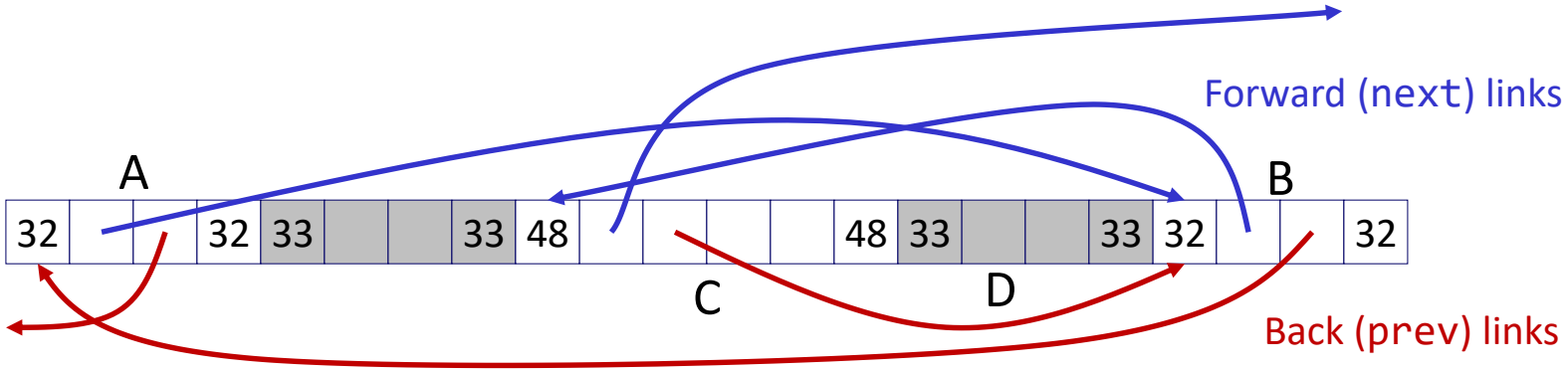


4) Return the address of the beginning of the payload

Deallocation: Constant Time Coalescing



Explicit List Summary



- Comparison with implicit list:
  - Block allocation is linear time in number of free blocks instead of all blocks
    - ***Much faster*** when most of the memory is full
  - Slightly more complicated allocate and free since we need to splice blocks in and out of the list
  - Some extra space for the links (2 extra pointers needed for each free block)
    - Increases minimum block size, leading to more internal fragmentation

Lesson Q&A

- Terminology:
  - Allocation strategies: first fit, next fit, best fit
  - Necessary block size, splitting, minimum block size, coalescing, boundary tags
  - Explicit free list (doubly-linked list)

- Learning Objectives:
  - Evaluate changes to the state of the heap for a sequence of allocations and deallocations.
  - Explain the tradeoffs between different allocator implementations, policies, and strategies.

- What lingering questions do you have from the lesson?

Memory Allocation II – Context

Allocation Policy Tradeoffs

- Data structure of blocks on lists
  - Implicit (free/allocated), explicit (free), segregated (many free lists) – others possible!
  - Metadata (*i.e.*, what tags we use in the boundary tags)
- Placement policy: first-fit, next-fit, best-fit
  - Throughput vs. amount of fragmentation
- When do we split free blocks?
  - How much internal fragmentation are we willing to tolerate?

Memory Allocation II – Practice

Practice Question

- Which allocation strategy and requests removes *external* fragmentation in this Heap? B3 was the last fulfilled request.

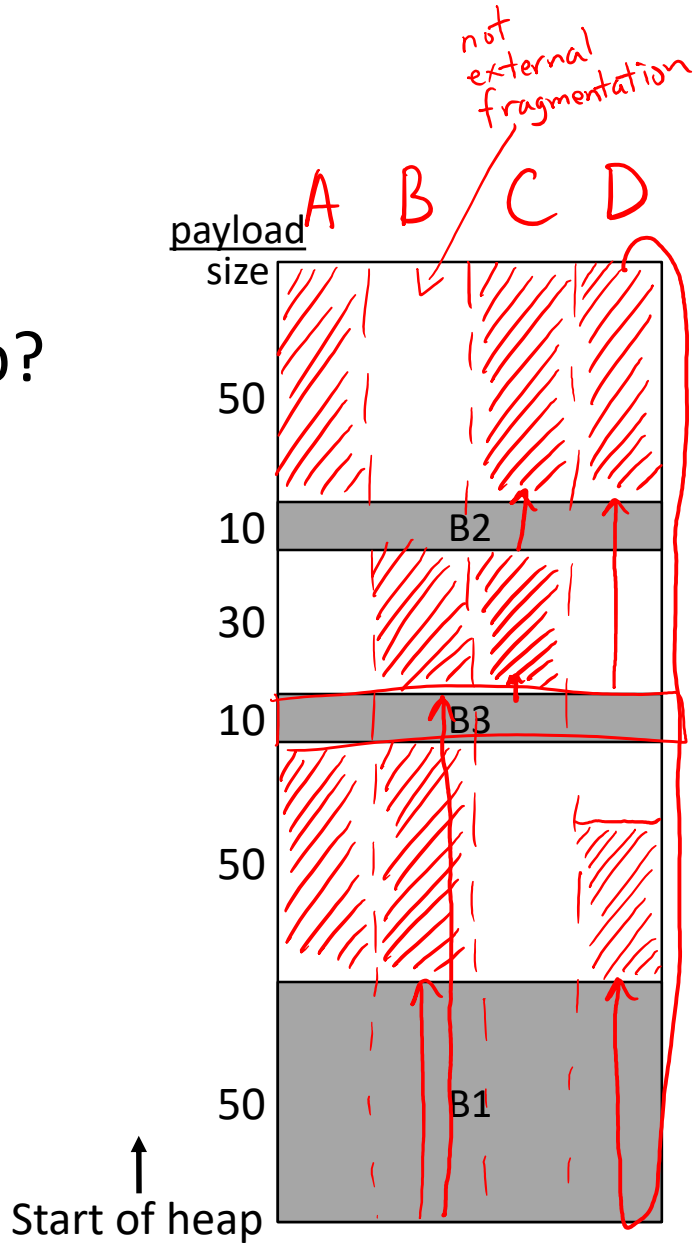
space between blocks

(A) Best-fit: `malloc(50), malloc(50)`

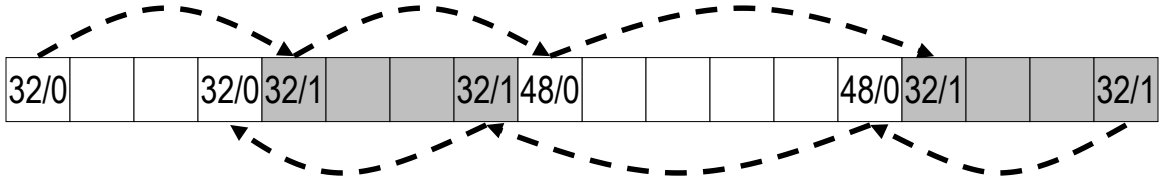
(B) First-fit: `malloc(50), malloc(30)`

(C) Next-fit: `malloc(30), malloc(50)`

(D) Next-fit: `malloc(50), malloc(30)`



Free List Review Questions



- What is the block header? What do we store and how?

  stores info about block

  size of block, is-allocated?
  ↑ lowest bit of header

- What are boundary tags and why do we need them?

  header and footer (same info)

  so we can traverse list in either direction (particularly for coalescing)

- When we coalesce free blocks, how many neighboring blocks do we need to check on either side? Why is this?

  just 1 — adjacent free blocks should have already been coalesced

Group Work Time

- During this time, you are encouraged to work on the following:
  1) If desired, continue your discussion
  2) Work on the homework problems
  3) Work on the current lab

- Resources:
  - You can revisit the lesson material
  - Work together in groups and help each other out
  - Course staff will circle around to provide support