CS130B- Data Structures and Algorithms II

Discussion Section Week 1

Thanks to
Emily Fujimoto & Fangqiu Han
for some slides

TAs

Name: Steven Munn
Email: sjm@ece.ucsb.edu
Office Hours:
MW 1pm - 3pm
F 2pm – 3pm

Name: Raymond Wong
Email: rwong@cs.ucsb.edu
Office Hours:
TW 1pm - 3pm
T 9:30am - 11:30am

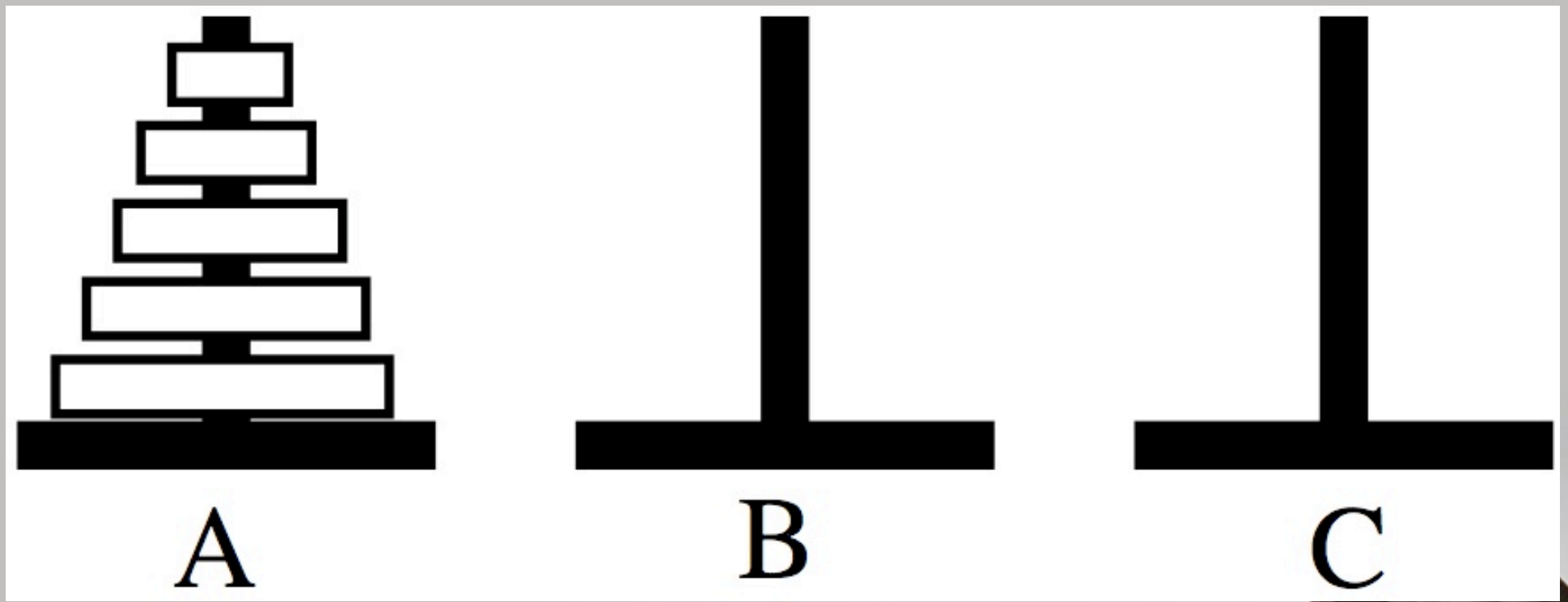
Course Website:
http://www.cs.ucsb.edu/~cs130b/

Class Email:
cs130b@cs.ucsb.edu

Written Assignment 1

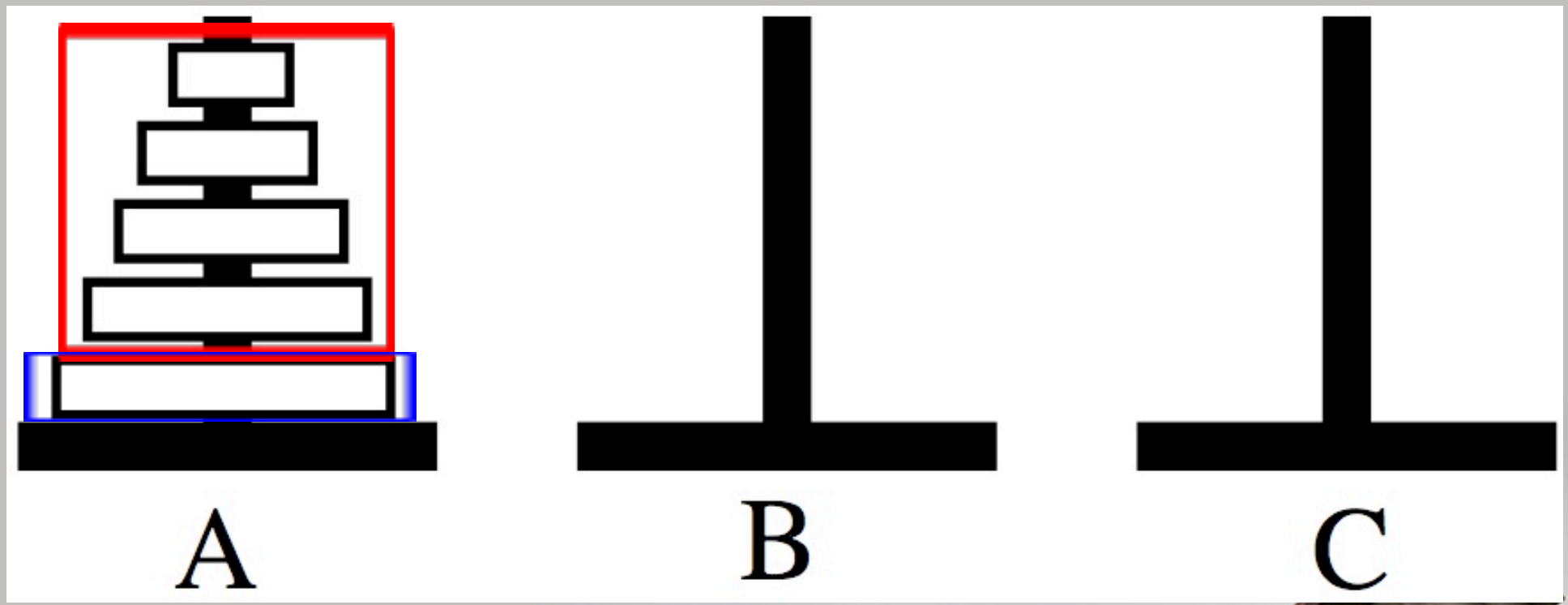
- Due Friday, April 14th at 4pm
- Turn in to CS130b box in the mail room (HFH 2108)
- 5 problems

Problem 1 – Tower of Hanoi



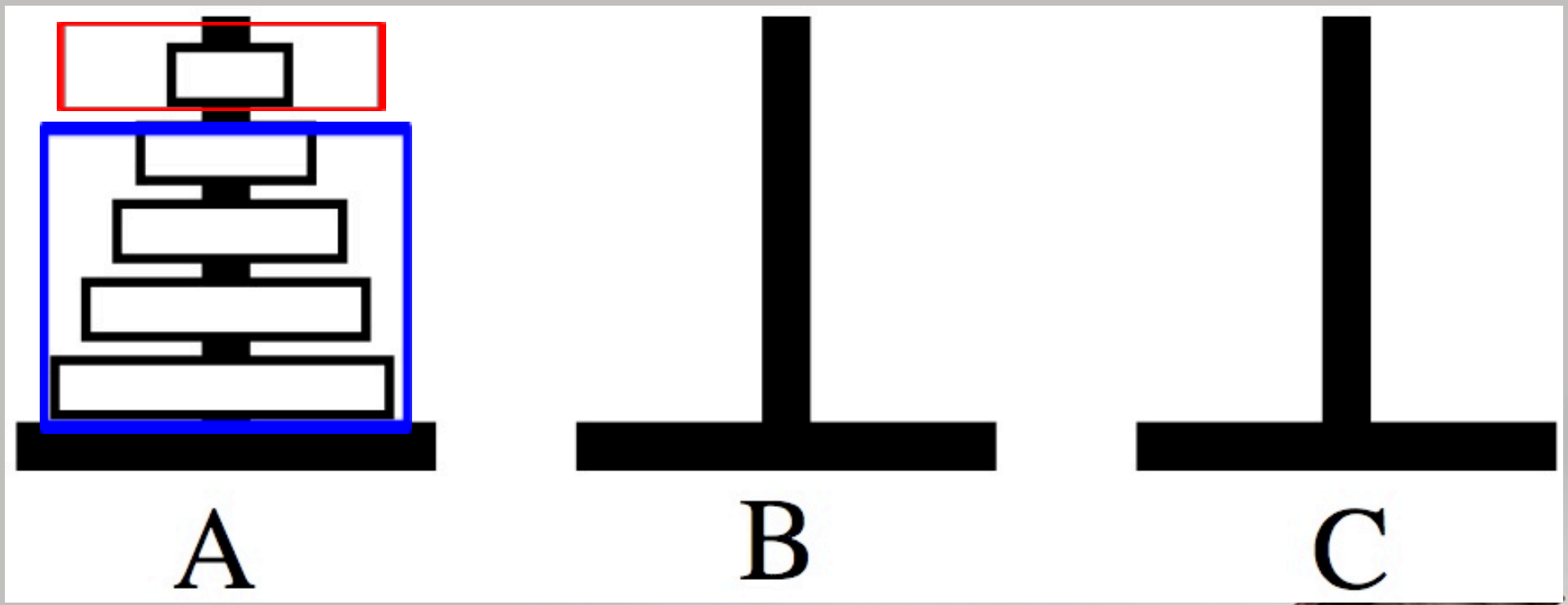
- Move disks from one peg to another
- Move one disk at a time
- Larger disks can't go on top of smaller ones

Problem 1 – Tower of Hanoi



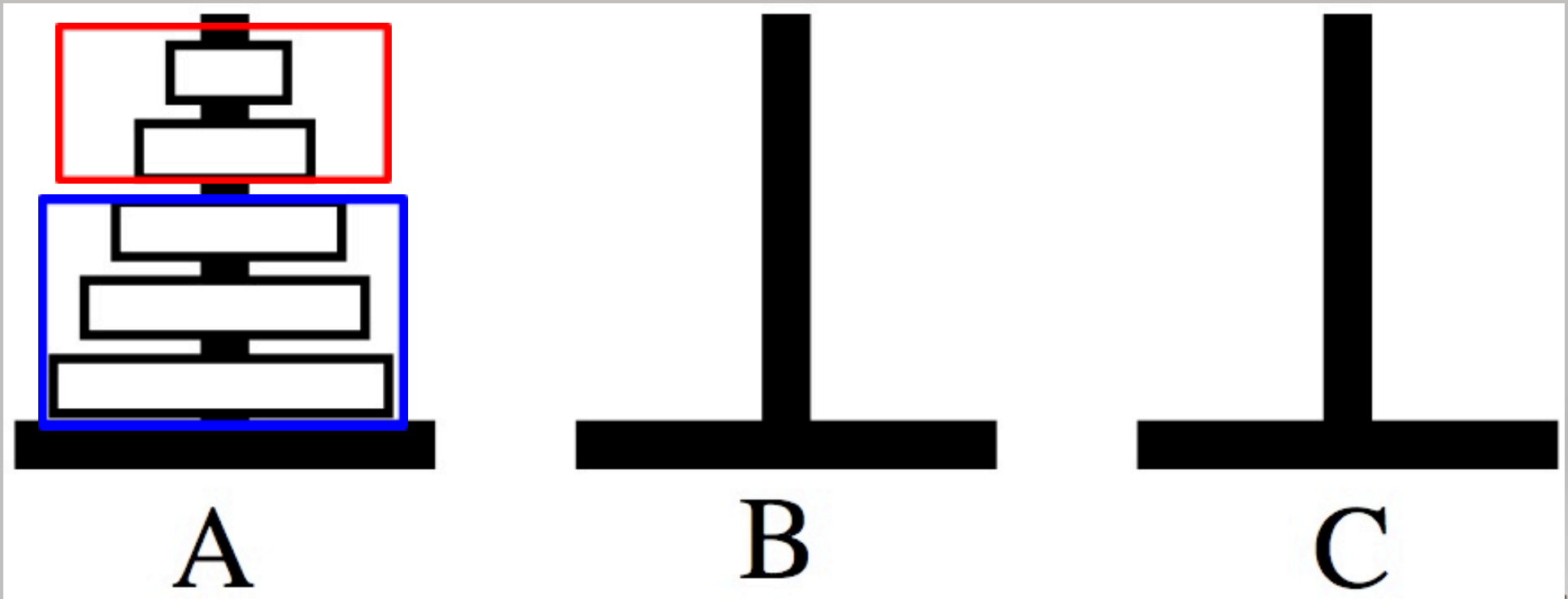
- In class solution:
  - Hanoi(n, A, B, C) = Hanoi(n-1, A, C, B) + Hanoi(1, A, B, C) + Hanoi(n-1, C, B, A)

Problem 1 – Tower of Hanoi



- Proposed Alternative 1:
  - Hanoi(n, A, B, C) = Hanoi(1, A, C, B) + Hanoi(n-1, A, B, C) + Hanoi(1, C, B, A)

Problem 1 – Tower of Hanoi



- Proposed Alternative 2:
  - Hanoi(n, A, B, C) = Hanoi(n/2, A, C, B) + Hanoi(n/2, A, B, C) + Hanoi(n/2, C, B, A)

Problem 2

Given a sorted array of n integers, find out if there exists index i such that $A[i] = i$ (assuming 1-based index)

Example:
[-17, -3, 1, 4, 6, 20]

We have $A[4] = 4$

Problem 2

Binary Search:
Want to find 42 in a sorted array

[1, 3, 5, 6, 9, 11, 14, 17, 21, 23, 27, 30, 42, 51, 55]

Problem 2

Binary Search:
Want to find 42 in a sorted array

[1, 3, 5, 6, 9, 11, 14, 17, 21, 23, 27, 30, 42, 51, 55]

[21, 23, 27, 30, 42, 51, 55]

Problem 2

Binary Search:
Want to find 42 in a sorted array

[1, 3, 5, 6, 9, 11, 14, 17, 21, 23, 27, 30, 42, 51, 55]

[21, 23, 27, 30, 42, 51, 55]

[42, 51, 55]

Problem 2

Binary Search:
Want to find 42 in a sorted array

[1, 3, 5, 6, 9, 11, 14, 17, 21, 23, 27, 30, 42, 51, 55]

[21, 23, 27, 30, 42, 51, 55]

[42, 51, 55]

[42]

Problem 3 – Finding the Median

Median in a sorted array:

[1, 2, 5, 7, 11, 15, 17, 21, 24]

Median in an unsorted array:

[42, 11, 32, 15, 1, 17, 7, 12, 37]

much harder to find

Problem 3 – Finding the Median

3a) Find a brute force solution.

3b) Find a Divide and Conquer Solution
Hint: How can we use the Quicksort algorithm to help?

Problem 3 – Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

Problem 3 – Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

[15, 12, 4, 16, 7 , 11, 42]

Go through array until you find values to swap

Problem 3 – Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

[15, 12, 4, 16, 7 , 11, 42]

Go through array until you find values to swap

[15, 12, 4, 16, 7 , 11, 42]

Problem 3 – Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

[15, 12, 4, 16, 7 , 11, 42]

Go through array until you find values to swap

[15, 12, 4, 16, 7 , 11, 42]

[15, 12, 4, 16, 7 , 11, 42]

Swap the values

Problem 3 – Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

[15, 12, 4, 16, 7 , 11, 42]

Go through array until you find values to swap

[15, 12, 4, 16, 7 , 11, 42]

[15, 12, 4, 16, 7 , 11, 42]

Swap the values

[15, 12, 4, 11, 7 , 16, 42]

Problem 3 – Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

[15, 12, 4, 16, 7 , 11, 42]

Go through array until you find values to swap

[15, 12, 4, 16, 7 , 11, 42]

Swap the values

[15, 12, 4, 16, 7 , 11, 42]

[15, 12, 4, 11, 7 , 16, 42]

[15, 12, 4, 11, 7 , 16, 42]

Problem 3- Finding the Median

Quicksort:

[15, 12, 4, 16, 7 , 11, 42]

Find the pivot

[15, 12, 4, 16, 7 , 11, 42]

Go through array until you find values to swap

[15, 12, 4, 16, 7 , 11, 42]

Swap the values

[15, 12, 4, 16, 7 , 11, 42]

[15, 12, 4, 11, 7 , 16, 42]

[7, 12, 4, 11, 15 , 16, 42]

Problem 4 – *n*-Way Merge

Given *n* sorted arrays, merge them into a single sorted array

4a) Naïve approach

First, merge the first two input arrays into one combined array, then merge this combined array with the third input array into an even bigger combined array, then merge this bigger combined array with the fourth array, and so on so forth. What's the complexity of this approach?

Problem 4 – n-Way Merge

4b) Divide-and-Conquer

Design a divide-and-conquer approach and analyze the complexity (assuming that n is the power of 2 for simplicity).

Hints:

Think about merge sort – is this essentially the same situation?

Image Sources

- http://pages.cs.brandeis.edu/~storer/JimPuzzles/MANIP/TowersOfHanoi/TowersOfHanoiFigure.jpg