

CSC 301: Analysis of Algorithms

Lecture 3: Designing Algorithms

William Rebelsky
[rebelskyw]

Grinnell College

Friday 26th January, 2024

Reminders

- Keep track of the Syllabus: Readings should be done before class not after
- Homework 0 is due Wednesday, you should have all the material to solve the problems
- We have a course website: `rebelskyw.cs.grinnell.edu`
- Tokens: Let me know if you have any events you would like me to add
- Quiz 0 has been released on gradescope and is due Monday
- Office Hours: 1604 Wednesday and Friday 1-2:50

Table of Contents

1 Practice Loop Invariants

- Red Blue Marbles
- Daisy Game

2 Merge Sort

Red Blue Marbles

Suppose you have a jar of one or more marbles, each of which is either RED or BLUE in color. You also have an unlimited supply of RED marbles off to the side. You then execute the following procedure: (example taken from Scranton)

Algorithm 1: RB-Marbles(m)

Input: Marbles**Output:** Color of Final Marble

```
1 while Number of marbles in the jar > 1 do
2   | choose (any) two marbles from the jar
3   | if the two marbles are of the same color then
4   |   | Remove them
5   |   | Place a RED marble in the jar
6   | else
7   |   | toss the chosen RED marble aside
8   |   | place the chosen BLUE marble back into the jar
```

Red Blue Marbles

- Can we guarantee that there is a single remaining marble?
- If we know the initial number of Red and Blue marbles, can we determine the color of the final marble?
- If so, prove it

BLANK

This frame intentionally left blank

Red Blue Marble Answers

- In each step the number of marbles decreases by exactly one
- Initially number of blue marbles is even or odd
- In each step of the loop, the number of blue marbles decreases by 0 or 2 so it remains even or odd
- At termination, there is one marble remaining. If blue is even, it must be red, if blue is odd, it must be blue.

Daisy Petal Game

- Imagine a Daisy with 16 petals
- Two players alternate turns
- In each turn, players must remove 1 or 2 adjacent petals (adjacent is based on the initial arrangement before any were removed)
- The winner is the player who takes the last petal
- Goal: Is there a way to guarantee that you win if you choose whether to go first or second?

BLANK

This frame intentionally left blank

Daisy Petal Game Answers

- The second player has a winning strategy of doing the opposite of the first player
- e.g. if the first player removes petal 1, remove petal 9
- Invariant: the second player has a move that can be made (if this is true, they haven't lost yet)
- Solution available at [The University of Texas at Austin](#)

Table of Contents

1 Practice Loop Invariants

- Red Blue Marbles
- Daisy Game

2 Merge Sort

Divide and Conquer

Recursion!

- **Divide** the problem into smaller instances of the same problem
- **Conquer** the subproblems by solving recursively
- **Combine** the subproblem solutions to form a solution to the original problem

Merge Sort

- What are the divide and conquer steps in merge sort?
- How long does merging take?
- What does the pseudo-code for Merge-Sort look like?
- Apply mergesort to the array [12,3,7,9,14,6,11,2]
- What is the running time of Merge-Sort utilizing a recursion tree?

BLANK

This frame intentionally left blank

Merge Sort Divide and Conquer

- **Divide** the subarray to be sorted into two adjacent arrays of half the size
- **Conquer** by sorting the subarrays recursively using merge sort
- **Combine** by merging the sorted subarrays producing the sorted answer. This is the key step: how do you combine?

Merge Sort: Merge time

- $\Theta(n)$

Merge Sort: Pseudo Code

Algorithm 2: Merge-Sort(A, p, r)

Input: sequence of n numbers $A = \{a_1, \dots, a_n\}$, left (p),
right (r)

Output: permutation $a'_1, \dots, a'_n$ such that $a'_1 \leq \dots \leq a'_n$

```
1 if  $p \geq r$  then  
2   | Return  
3  $q = \lfloor (p + r) / 2 \rfloor$   
4 Merge-Sort( $A, p, q$ )  
5 Merge-Sort( $A, q+1, r$ )  
6 Merge( $A, p, q, r$ )
```

Merge Sort: Running time

$$T(n) = \begin{cases} \Theta(1) & n < n_0, \\ D(n) + aT(n/b) + C(n) & \text{otherwise} \end{cases}$$

- Assume n is a power of 2
- Draw the tree: Left to the reader
- Analyze the tree: height $\log n + 1$, each row is $c_2 n$ except the last which is $c_1 n = c_2 n \log n + c_1$
- $\Theta(n \log n)$