

My Notes 2024

Department of Computer Science, Copenhagen University (DIKU)
Algorithms and Data Structures

Simon Winther
zlp616@alumni.ku.dk

November 27, 2023.

Contents

1	5/2: Introduction, Proofs, Loop Invariants, CLRS 2.1 (RP)	1
1.1	Loop Invariants:	1
1.2	Proofs:	1
1.3	Insertion-Sort and cost analysis	2
1.4	Exercises:	3
2	7/2: Time Complexity and Asymptotic Notation, CLRS 2.2, 3 (RP)	5
3	12/2: Divide and Conquer, Recursion Equations, and Lower Bound for Sorting, CLRS 2.3, 4 up to 4.5, pages 157-160, 8.1 (RP)	5
4	14/2: Amortized Analysis, CLRS 16 (RP)	5
5	19/2: LSM Trees, Fibonacci Heaps, CLRS digital chapter 19 Download digital chapter 19 (19.4 is cursory) (RP)	5
6	21/2: Dynamic Programming, CLRS 14 except for 14.2 and 14.5 (MA)	5
7	26/2: Greedy Algorithms, CLRS 15 (MA)	5
8	28/2: No teaching due to open house	5
9	4/3: Binary Search Trees, CLRS 12 and 13 (PW)	5
10	6/3: Disjoint Sets, Plane Sweep, CLRS 19.1-3, CLRS digital chapter 33.1-2 Download CLRS digital chapter 33.1-2 (PW)	5
11	11/3: Minimum Spanning Tree, CLRS 21 (PW)	5
12	13/3: Shortest Paths, CLRS 22 (PW)	5

13 18/3: Computational Geometry, CLRS digital chapter 33 Download CLRS digital chapter 33 (PW)	5
14 4/3: Question Time (same time and place as lectures) (MA, PW, RP)	5
15 8/4 OR 10/4?: Written Exam	5
16 Appendix	5

List of Algorithms

1	Insertion-Sort(A)	2
---	-----------------------------	---

!So, writing should be done without ur wrists touching the table or the wrist pad. This is to avoid RSI.

1 5/2: Introduction, Proofs, Loop Invariants, CLRS 2.1 (RP)

Chapter Quick Summary Notes (CLRS 2.1)

This box can contain summaries, key points, and important definitions from the reading material assigned (CLRS 2.1). Highlight the main concepts, proofs, and examples discussed in the text.

1.1 Loop Invariants:

Loop invariants are used to prove the correctness of algorithms. They are used to show that the algorithm is working as expected. The loop invariant is a property that holds before and after each iteration of the loop. When you're using a loop invariant, you need to show three things:

1. **Initialization:** It is true prior to the first iteration of the loop.
2. **Maintenance:** If it is true before an iteration of the loop, it remains true before the next iteration.
3. **Termination:** The loop terminates, and when it terminates, the invariant - usually along with the reason that the loop terminated - gives us a useful property that helps show that the algorithm is correct.

When the first two properties are true, the loop invariant is true prior to every iteration of the loop. When the third property is true, the loop invariant is true after the last iteration of the loop, giving us a useful property that helps show that the algorithm is correct.

A loop-invariant proof is a form of mathematical induction, where to prove that a property holds, you prove a base case and an inductive step. Here, showing the invariant holds before the first iteration of the loop is like the base case, and showing that the invariant holds from one iteration to the next (from iteration to iteration) is like the inductive step. The third property is perhaps the most important one, because it is where the loop invariant is used to show that the algorithm is correct.

1.2 Proofs:

Test.

1.3 Insertion-Sort and cost analysis

Algorithm 1 Insertion-Sort(A)	cost	times
Input: $A[1..n]$, an array of numbers.		
Output: $A[1..n]$, sorted in non-decreasing order.		
1: for $j = 2$ to $A.length$ do	c_1	n
2: $key = A[j]$	c_2	$n - 1$
3: $i = j - 1$	0	$n - 1$
4: while $i > 0$ and $A[i] > key$	c_4	$n - 1$
do		
5: $A[i + 1] = A[i]$	c_5	$\sum_{j=2}^n t_j$
6: $i = i - 1$	c_6	$\sum_{j=2}^n (t_j - 1)$
7: $A[i + 1] = key$	c_7	$\sum_{j=2}^n (t_j - 1)$

where t_j is the number of times the while loop test in line 4 is executed for that value of j . $T(n)$ also called the recurrence relation, is given by:

$$\begin{aligned}
 T(n) &= c_1 n + c_2(n - 1) + c_4(n - 1) + c_5 \sum_{j=2}^n t_j + c_6 \sum_{j=2}^n (t_j - 1) + c_7 \sum_{j=2}^n (t_j - 1) \\
 &= (c_1 + c_2 + c_4)n - (c_2 + c_4 + c_5 + c_6 + c_7) + (c_5 + c_6 + c_7) \sum_{j=2}^n (t_j - 1)
 \end{aligned}$$

For the worst case we have $t_j = j$ for $j = 2, 3, \dots, n$ because each j iteration it has to move the key to the first position and we know that $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ and so hence we would have $T(n)$ as follows:

$$\begin{aligned}
 T(n) &= (c_1 + c_2 + c_4)n - (c_2 + c_4 + c_5 + c_6 + c_7) + (c_5 + c_6 + c_7) \sum_{j=2}^n (j - 1) \\
 &= (c_1 + c_2 + c_4)n - (c_2 + c_4 + c_5 + c_6 + c_7) + (c_5 + c_6 + c_7) \left(\frac{n(n-1)}{2} - (n-1) \right) \\
 &= an^2 + bn + c
 \end{aligned}$$

And thereby we have shown that the worst-case running time of insertion-sort is $\Theta(n^2)$.

1.4 Exercises:

Question 1

Prove that $x^2 - y^2 = 1$ has no solutions for positive integers x, y (the positive integers are the numbers $1, 2, \dots$).

Proof for Question 1: We can consider the equation in the form of difference of two squares:

$$x^2 - y^2 = (x - y)(x + y) = 1$$

We can now recall that for positive integers a and b , if $ab = 1$, then both a and b must be 1. Since 1 is the only positive integer multiplied by itself that gives 1. Hence, when we consider the equation $(x - y)(x + y) = 1$, we can see that the only way to get 1 is if $x - y = 1$ and $x + y = 1$. This gives us the system of equations:

$$\begin{cases} x - y = 1 \\ x + y = 1 \end{cases}$$

Adding the two equations gives us $2x + 0y = 2x = 2$, and so $x = 1$. Substituting $x = 1$ into the second equation gives us $1 + y = 1$, and so $y = 0$. But 0 is not a positive integer, and so the equation $x^2 - y^2 = 1$ has no solutions for positive integers x, y . \square

Question 2: Question 2

Prove that $x^2 - y^2 = 10$ has no solutions for positive integers x, y

Proof for Question 2: Again, we can consider the equation in the form of difference of two squares:

$$x^2 - y^2 = (x - y)(x + y) = 10$$

This time we will in fact assume that $\exists x, y \in \mathbb{Z}^+$ such that $x^2 - y^2 = 10$. If we consider the equation $(x - y)(x + y) = 10$, \square

Question 3

Prove that if a is a rational number and b is an irrational number, then $a + b$ is an irrational number. (A rational number is one that can be written in the form $\frac{m}{n}$, where m and n are integers and $n \neq 0$).

Question 4

Give a formula for the sum of the first n odd numbers.

$$f(n) = \sum_{i=1}^n (2i - 1)$$

Question 5

Show that $n! > 2^n$ for $n \geq 4$.

Question 6

The Fibonacci numbers are defined as $f_0 = 0, f_1 = 1$, and $f_n = f_{n-1} + f_{n-2}$, for $n \geq 2$. The first Fibonacci numbers are thus $0, 1, 1, 2, 3, 5, 8, 13, 21, \dots$. Prove that

$$\sum_{i=1}^n f_i^2 = f_n \cdot f_{n+1}$$

Question 7

Show that

$$\sum_{i=1}^n f_{2i} = f_{2n} - 1$$

Question 8

Show that

$$\sum_{i=1}^n \frac{1}{i(i+1)} = \frac{n}{n+1}$$

- 2 7/2: Time Complexity and Asymptotic Notation, CLRS 2.2, 3 (RP)
- 3 12/2: Divide and Conquer, Recursion Equations, and Lower Bound for Sorting, CLRS 2.3, 4 up to 4.5, pages 157-160, 8.1 (RP)
- 4 14/2: Amortized Analysis, CLRS 16 (RP)
- 5 19/2: LSM Trees, Fibonacci Heaps, CLRS digital chapter 19 Download digital chapter 19 (19.4 is cursory) (RP)
- 6 21/2: Dynamic Programming, CLRS 14 except for 14.2 and 14.5 (MA)
- 7 26/2: Greedy Algorithms, CLRS 15 (MA)
- 8 28/2: No teaching due to open house
- 9 4/3: Binary Search Trees, CLRS 12 and 13 (PW)
- 10 6/3: Disjoint Sets, Plane Sweep, CLRS 19.1-3, CLRS digital chapter 33.1-2 Download CLRS digital chapter 33.1-2 (PW)
- 11 11/3: Minimum Spanning Tree, CLRS 21 (PW)
- 12 13/3: Shortest Paths, CLRS 22 (PW)
- 13 18/3: Computational Geometry, CLRS digital chapter 33 Download CLRS digital chapter 33 (PW)
- 14 4/3: Question Time (same time and place as lectures) (MA, PW, RP)
- 15 8/4 OR 10/4?: Written Exam
- 16 Appendix