DIVISION OF DIGITAL CONTENTS
DONGSEO UNIVERSITY

Algorithm

Time Complexity

jintaeks@dongseo.ac.kr

December 3rd, 2018

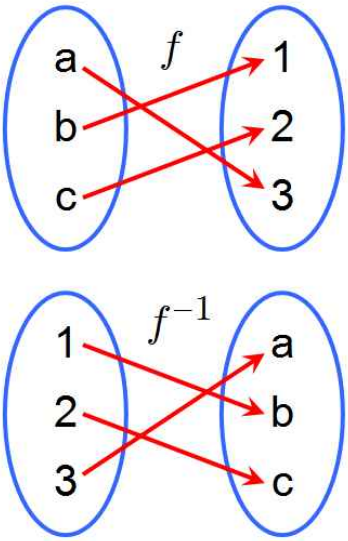
DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

The elementary functions

- ✓ Powers of x : x, x^2, x^3, etc.
- ✓ Roots of x : $\sqrt{x}$,$\sqrt[3]{x}$ etc.
- ✓ Exponential functions: e^x
- ✓ Logarithms: log x
- ✓ Trigonometric functions: sin(x), cos(x) etc.
- ✓ Inverse trigonometric functions
- ✓ Hyperbolic functions: sinh(x), cosh(x) etc.
- ✓ All functions obtained by replacing x with any of the previous functions
- ✓ All functions obtained by adding, subtracting, multiplying or dividing any of the previous functions

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Inverse function



✓ In mathematics, an **inverse function** is a function that "reverses" another function. That is, if *f* is a function mapping *x* to *y*, then the inverse function of *f* maps *y* back to *x*.

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Function $f(x)$	Inverse $f^{-1}(y)$	Notes
$x+a$	$y-a$	
$a-x$	$a-y$	
mx	$\frac{y}{m}$	$m \neq 0$
$\frac{1}{x}$ (i.e. x^{-1})	$\frac{1}{y}$ (i.e. y^{-1})	$x, y \neq 0$
x^2	$\sqrt{y}$ (i.e. $y^{1/2}$)	$x, y \geq 0$ only
x^3	$\sqrt[3]{y}$ (i.e. $y^{1/3}$)	no restriction on x and y
x^p	$\sqrt[p]{y}$ (i.e. $y^{1/p}$)	$x, y \geq 0$ in general, $p \neq 0$
2^x	$\mathrm{lb} y$	$y > 0$
e^x	$\ln y$	$y > 0$
10^x	$\lg y$	$y > 0$
a^x	$\log_a y$	$y > 0$ and $a > 0$
trigonometric functions	inverse trigonometric functions	various restrictions (see table below)
hyperbolic functions	inverse hyperbolic functions	various restrictions

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Square Root

- ✓ In mathematics, a **square root** of a number *a* is a number *y* such that $y^2 = a$, in other words, a number *y* whose *square*(the result of multiplying the number by itself, or $y \times y$) is *a*.
- ✓ For example, 4 and −4 are square roots of 16 because $4^2 = (-4)^2 = 16$.

$$\sqrt[2]{16} = 4$$

5

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Cubic Root

- ✓ A Root is a general notation for all degrees.

$$\sqrt[3]{27} = 3$$

- ✓ In case of square root, we can omit superscript 2.

$$\sqrt{16} = 4$$

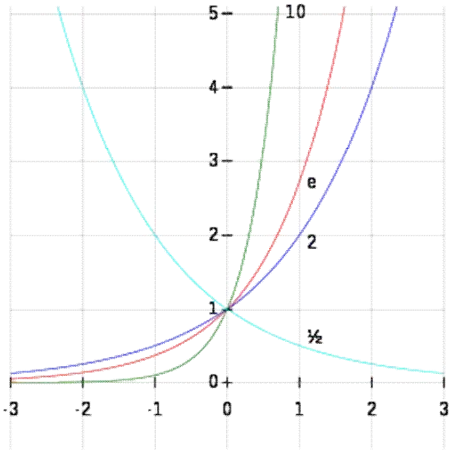
6

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Exponentiation

- ✓ **Exponentiation** is a mathematical operation, written as b^n, involving two numbers, the **base** b and the **exponent** n.
- ✓ When n is a positive integer, exponentiation corresponds to repeated multiplication of the base: that is, b^n is the product of multiplying n bases:

$$b^n = \underbrace{b \times \cdots \times b}_{n}$$



7

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Logarithm

- ✓ In mathematics, the **logarithm** is the inverse operation to exponentiation.
- ✓ That means the logarithm of a number is the exponent to which another fixed value, the base, must be raised to produce that number.
- ✓ In simple cases the logarithm counts repeated multiplication.
- ✓ For example, the base 10 logarithm of 1000 is 3, as 10 to the power 3 is 1000 ($1000 = 10 \times 10 \times 10 = 10^3$); the multiplication is repeated three times.

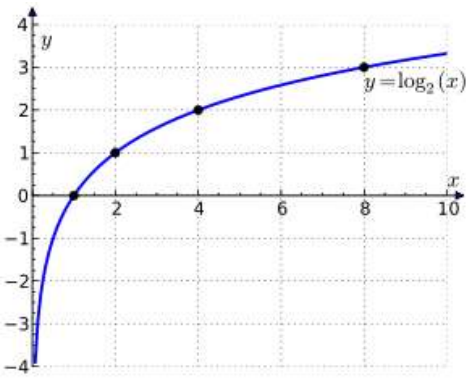
8

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

- ✓ The **logarithm of *x* to *base b*(log *base b* of *x*)**, denoted $\log_b(x)$, is the unique real number y such that $b^y = x$.
- ✓ For example, as $64 = 2^6$, we have $\log_2(64) = 6$

9

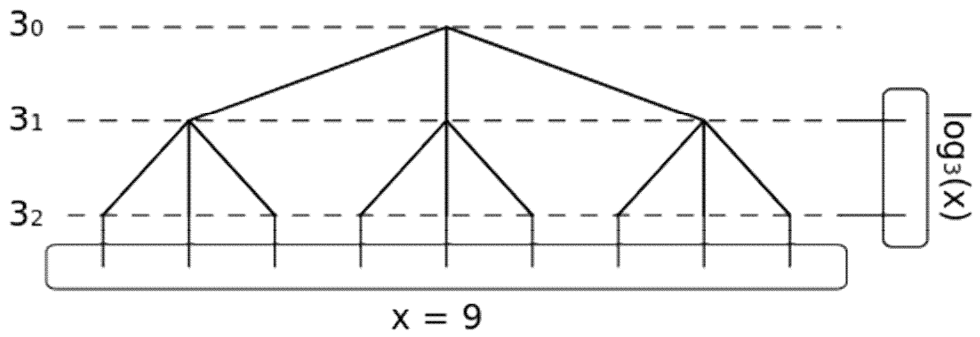
DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY



- ✓ The graph of the logarithm to base 2 crosses the *x* axis (horizontal axis) at 1 and passes through the points with coordinates (2, 1), (4, 2), and (8, 3). For example, $\log_2(8) = 3$, because $2^3 = 8$. The graph gets arbitrarily close to the *y* axis, but does not meet or intersect it.

10

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY



✓ A full 3-ary tree can be used to visualize the exponents of 3 and how the logarithm function relates to them.

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Product, quotient, power and root

	Formula	Example
product	$\log_b(xy) = \log_b(x) + \log_b(y)$	$\log_3(243) = \log_3(9 \cdot 27) = \log_3(9) + \log_3(27) = 2 + 3 = 5$
quotient	$\log_b\left(\frac{x}{y}\right) = \log_b(x) - \log_b(y)$	$\log_2(16) = \log_2\left(\frac{64}{4}\right) = \log_2(64) - \log_2(4) = 6 - 2 = 4$
power	$\log_b(x^p) = p\log_b(x)$	$\log_2(64) = \log_2(2^6) = 6\log_2(2) = 6$
root	$\log_b \sqrt[p]{x} = \frac{\log_b(x)}{p}$	$\log_{10}\sqrt{1000} = \frac{1}{2}\log_{10} 1000 = \frac{3}{2} = 1.5$

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

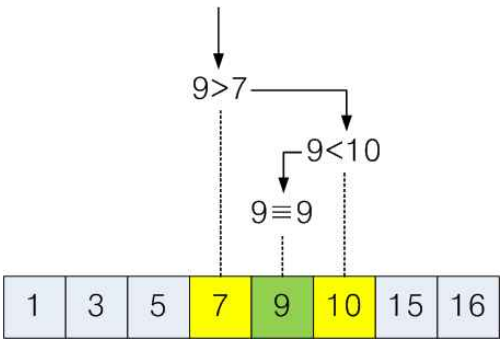
Linear Search

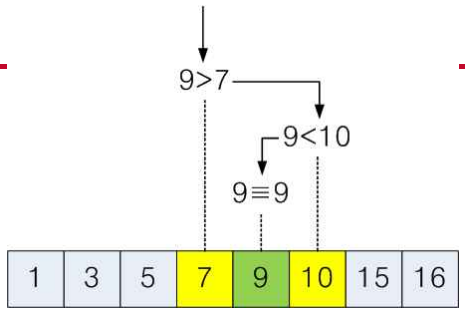
1. Set i to 0.
2. If $L_i \geq T$, go to step 4.
3. Increase i by 1 and go to step 2.
4. If $L_i = T$, the search terminates successfully; return i. Else, the search terminates unsuccessfully.

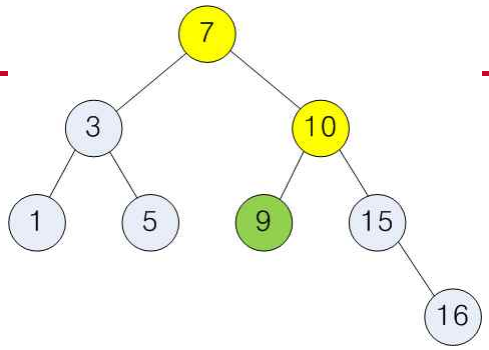
$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

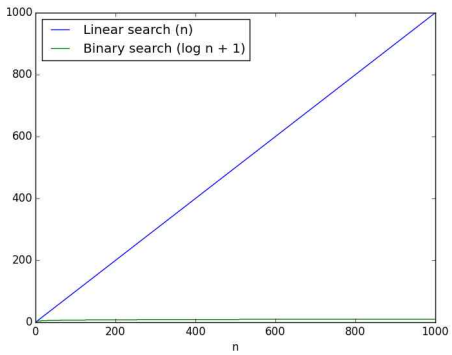
Binary search

✓ In computer science, **binary search**, also known as **half-interval search** or **logarithmic search**,[1] is a search algorithm that finds the position of a target value within a sorted array.









: Number of iterations for binary and linear search for n from 1 to 1000.

15

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Time complexity

- ✓ In computer science, the **time complexity** of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the string representing the input.
- ✓ The time complexity of an algorithm is commonly expressed using big O notation, which excludes coefficients and lower order terms.
- ✓ When expressed this way, the time complexity is said to be described ***asymptotically***, i.e., as the input size goes to infinity.
- ✓ For example, if the time required by an algorithm on all inputs of size n is at most $5n^3 + 3n$ for any n (bigger than some n_0), the asymptotic time complexity is $O(n^3)$.

16

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

Sort Algorithms

Name	Best	Average	Worst	Memory	Stable	Method
Quicksort	$n \log n$ variation is n	$n \log n$	n^2	$\log n$ on average, worst case is n; Sedgewick variation is $\log n$ worst case	typical in-place sort is not stable; stable versions exist	Partitioning
Merge sort	$n \log n$	$n \log n$	$n \log n$	n	Yes	Merging
In-place merge sort	—	—	$n \log^2 n$	1	Yes	Merging
Heapsort	$n \log n$	$n \log n$	$n \log n$	1	No	Selection
Insertion sort	n	n^2	n^2	1	Yes	Insertion

Bubble Sort

```
procedure bubbleSort( A : list of sortable items )
    n = length(A)
    repeat
        swapped = false
        for i = 1 to n-1 inclusive do
            /* if this pair is out of order */
            if A[i-1] > A[i] then
                /* swap them and remember something changed */
                swap( A[i-1], A[i] )
                swapped = true
            end if
        end for
    until not swapped
end procedure
```

Quick sort

- ✓ Quicksort is a fast sorting algorithm, which is used not only for educational purposes, but widely applied in practice.
- ✓ On the average, it has O(n log n) complexity, making quicksort suitable for sorting big data volumes.

Algorithm

- ✓ The divide-and-conquer strategy is used in quicksort. Below the recursion step is described:

1. Choose a pivot value. We take the value of the middle element as pivot value, but it can be any value, which is in range of sorted values, even if it doesn't present in the array.

2. Partition. Rearrange elements in such a way, that all elements which are lesser than the pivot go to the left part of the array and all elements greater than the pivot, go to the right part of the array. Values equal to the pivot can stay in any part of the array. Notice, that array may be divided in non-equal parts.

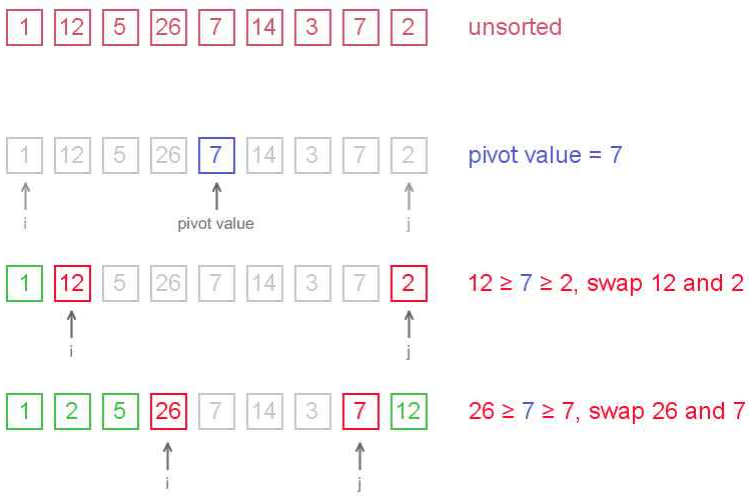
3. Sort both parts. Apply quicksort algorithm recursively to the left and the right parts.

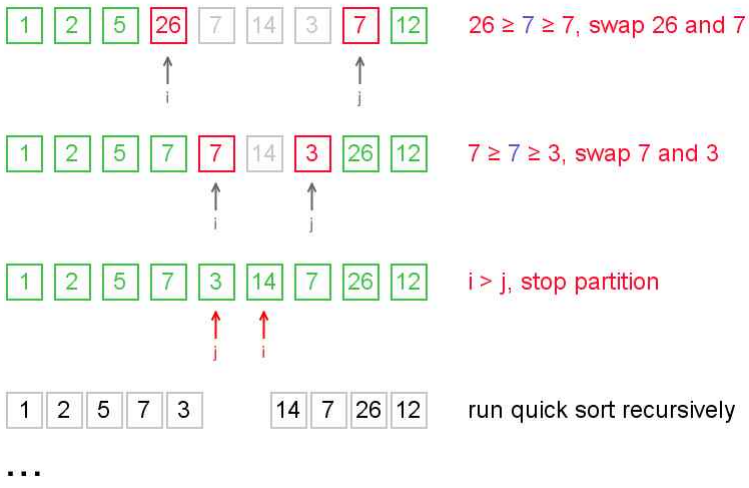
Partition algorithm in detail

- ✓ There are two indices **i** and **j** and at the very beginning of the partition algorithm **i** points to the first element in the array and **j** points to the last one.
- ✓ Then algorithm moves **i** forward, until an element with value greater or equal to the pivot is found.
- ✓ Index **j** is moved backward, until an element with value lesser or equal to the pivot is found.
- ✓ If **i ≤ j** then they are swapped and i steps to the next position (**i + 1**), j steps to the previous one **(j - 1)**.
- ✓ Algorithm stops, when **i** becomes greater than **j**.
- ✓ After partition, all values before **i-th** element are less or equal than the pivot and all values after **j-th** element are greater or equal to the pivot.

example

- ✓ Sort {1, 12, 5, 26, 7, 14, 3, 7, 2} using quicksort.





DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY

QnA

MY BRIGHT FUTURE

DSU Dongseo University 동서대학교

DIVISION OF DIGITAL CONTENTS DONGSEO UNIVERSITY