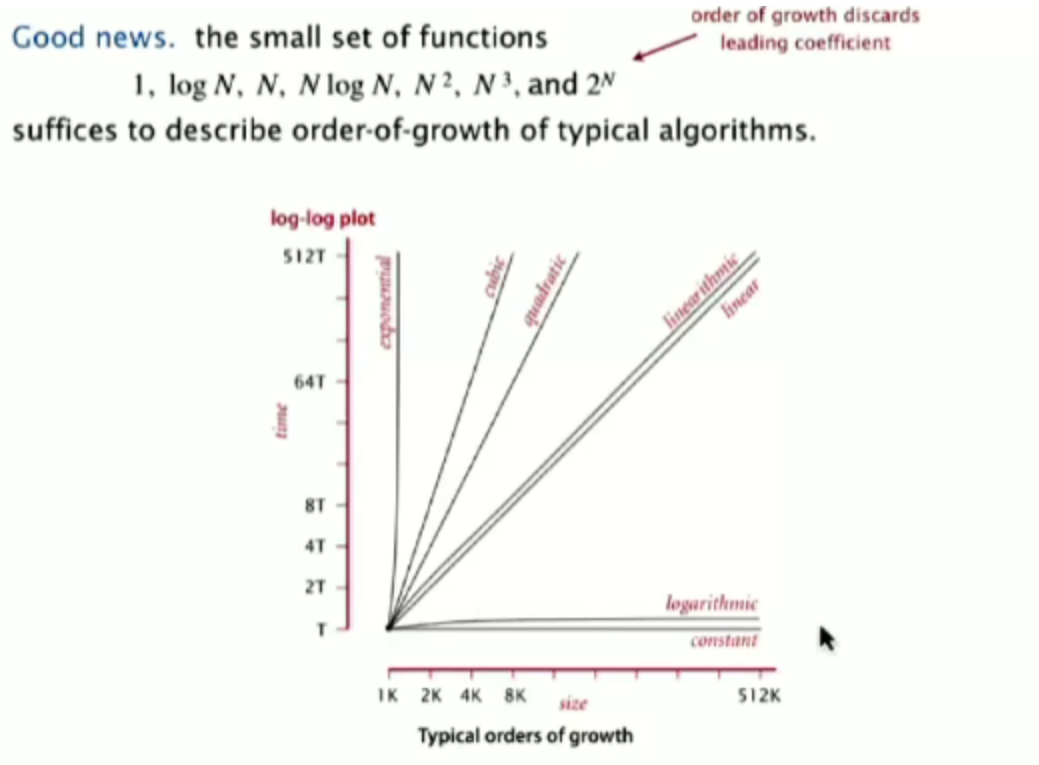
【DS】Day3

Tags	
Date	@May 22, 2022
Summary	Analysis of Algorithms

【Week1】Analysis of Algorithms

1.8 Order-of-Growth Classification

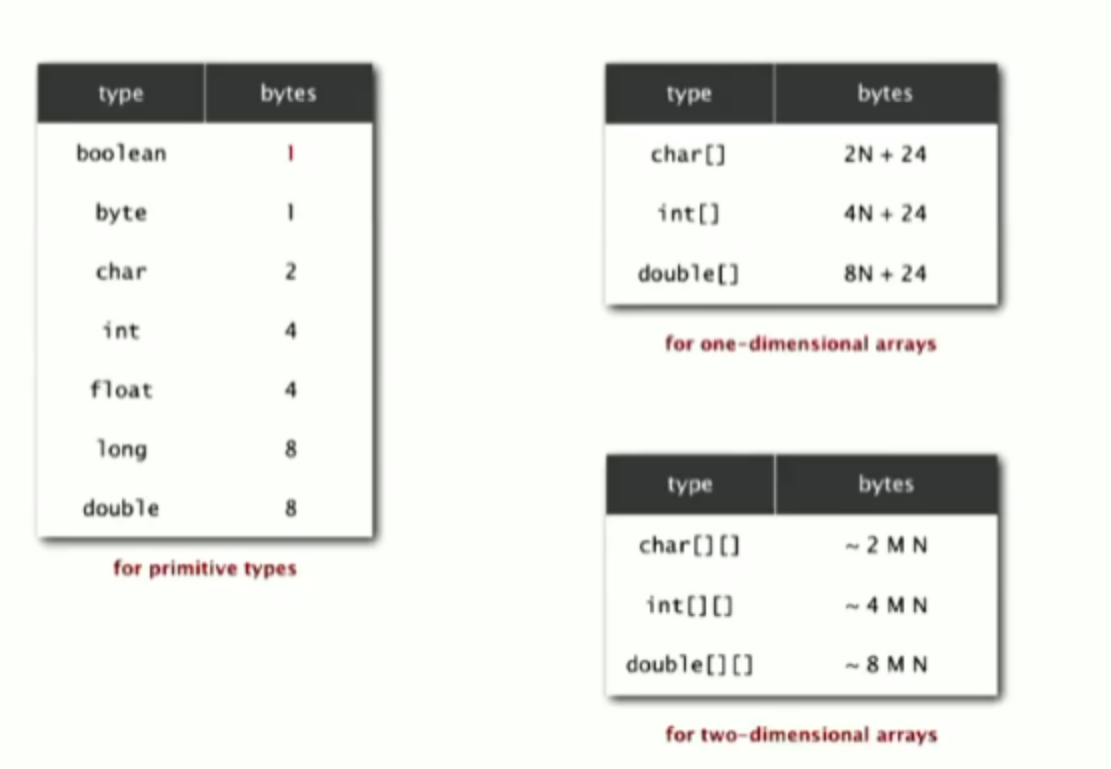
Common order-of-growth classification



order of growth	name	typical code framework	description	example	T(2N) / T(N)
1	constant	`a = b + c;`	statement	add two numbers	1
log N	logarithmic	`while (N > 1) { N = N / 2; ... }`	divide in half	binary search	~ 1
N	linear	`for (int i = 0; i < N; i++) { ... }`	loop	find the maximum	2
N log N	linearithmic	[see mergesort lecture]	divide and conquer	mergesort	~ 2
N^2	quadratic	`for (int i = 0; i < N; i++) for (int j = 0; j < N; j++) { ... }`	double loop	check all pairs	4
N^3	cubic	`for (int i = 0; i < N; i++) for (int j = 0; j < N; j++) for (int k = 0; k < N; k++) { ... }`	triple loop	check all triples	8
2^N	exponential	[see combinatorial search lecture]	exhaustive search	check all subsets	T(N)

1.9 Memory

Modern Machine: We assume a 64-bit machine with 8 byte pointers.



type	bytes
boolean	1
byte	1
char	2
int	4
float	4
long	8
double	8

for primitive types

type	bytes
char[]	2N + 24
int[]	4N + 24
double[]	8N + 24

for one-dimensional arrays

type	bytes
char[][]	~ 2 M N
int[][]	~ 4 M N
double[][]	~ 8 M N

for two-dimensional arrays

Object overhead: 16 bytes

Reference: 8 bytes

Padding: Each object uses a multiple of 8 bytes

Example: A virgin String of length N uses ~2N bytes of memory

```
// 16 bytes(object overhead)
// 8 bytes reference to array
// 2N + 24 bytes(char[] arra)
// 3 * 4-byte ints
// 4 byte padding
public class String {
  private char[] value;
  private int offset;
  private int count;
  private int hash;
}
```