Analysis of Algorithms

- Why is my program so slow?
- Why does it run out of memory?

three-sum problem → how many unique triplets on an array when summed result is zero

- brute force - N^3
- sort, get 2 numbers
  get the 3rd w/ binary search $N^2 \log N$

Experimental Observations

- measure running time manually (stopwatch)
  1. plot running time as function of # inputs ($T(N)$ vs N
  2. plot loglog plot do → linear regression $mx+b$

If $\lg(T(N) = b \log N + c$

then $t(N) = a N^B$ where $a = 2^c$

$t(N) = 2^c \cdot N^B$

- Recommended book = Art of computer programming
  Donald Knuth

System Dependent Effects } depends on computer specifications and lowlevel implementation

Cost of Arithmetic operation

- integer operations faster than float operations
- divide operations slower than Addition
- sine and arctangent are significantly slower than operations above

BASIC operations

Constant

- declaring variables
- Assignment
- integer compare
- Array element Access
- Array length
  String ↗

dependent on size

- Array Allocation
- String concatenation

Examples of counting BASIC operations (instructions) based on input size N

- finding how many zeros on An Array
- finding 2-sum problem
- finding 3-sum problem

→ Check Slides

ORDER OF GROWTH CLASSIFICATION

- Important!

ORDER OF GROWTH CLASSIFICATIONS

1
log N
N
N log N
N^2
$N^2 \log N$
N^3
2^N

* plot the Graphs of theses

→ plog Log Log plot also.

AMORTIZED

ORDER OF GROWTH

- See Stack Implementation

LOGARITHMIC

- BINARY SEARCH • divide in half
- while N > 1:
  N = N/2

NLogN (LINEARITHMIC)

- merge sort

→ divide and conquer

2^N EXPONENTIAL

- Exhaustive search
  Check all subsets
  COMBINATORIAL SEARCH

64 bit MACHINE vs 32bit

- 8 byte pointers
- more memory
- more pointers possible

MEMORY

PRIMITIVE TYPES

- boolean • int
- byte • float
- CHAR • long

OBJECTS

- object overhead
- reference
- padding.

1D/2D ARRAYS

- depends of size of what's inside each cell

Example:

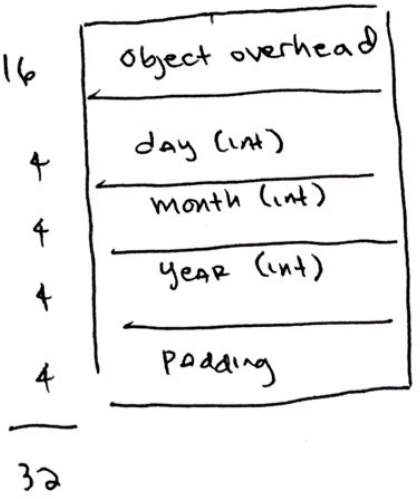
CHAR [][] → c N·M + padding

- SIZEOF CHAR
- NROWS
- MCOLUMS
- PADDING theader

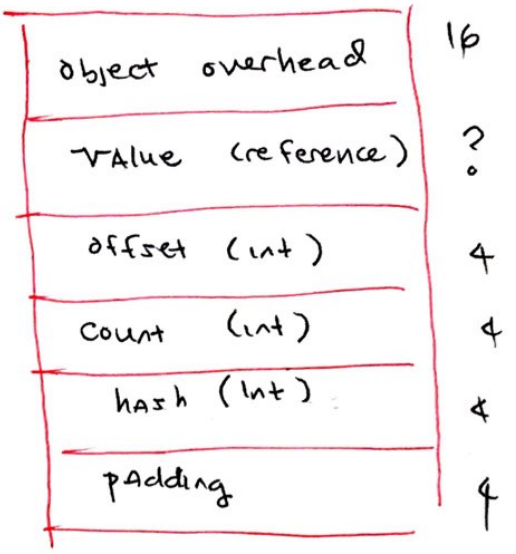
usage

MEMORY, CONTINUED

- see Example class Date



class String



SHALLOW MEMORY

- dont count referenced objects (pointed to)

vs deep memory

value reference

8 bytes (pointer reference to Array)

2N + 24 (char Array)

- # of bytes per char
- # characters
- overhead

Example Weighted Quick Union memory

16 → object overhead

→ Arrays (id), (size)

4 → int count

4 → padding

F

→ 8 + 4N + 24

EACH

memory = 8N + 88

THings to DO

- → Implement binary search
- → Implement 2 Algorithms for 3-sum problem
- → plot benchmarks for 2 implementations of 3-sum
- → plot order of growth graph
- → scan notes