Finding the Median under IOI Conditions

Tom Verhoeff

Technische Universiteit Eindhoven
Faculteit Wiskunde en Informatica
Software Constructie Groep

`T.Verhoeff@TUE.NL`
`http://www.win.tue.nl/~wstomv/`

Joint work with *Gyula Horváth*, University of Szeged, Hungary

Overview

- Programming Contests
- IOI 2000 Task *Median*
- Invitation

Programming Contests

- ACM ICPC – International Collegiate Programming Contest
- IOI – International Olympiad in Informatics

ACM ICPC versus IOI

ACM ICPC	IOI
for university/college students	for high-school students
for teams of 3	for individuals
satisfy the jury	satisfy the specification
judged during contest	judged after contest
pass/fail judging	graded scores
winner takes it all	top 50% receive medals

IOI 2000: Palindrome

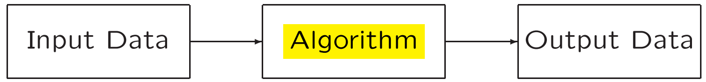
Informal specification:

> Design an efficient algorithm that
>
> reads a sequence of characters and
>
> writes the minimum number of characters to be
>
> inserted into the input sequence to yield a palindrome.

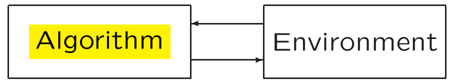
Input `Ab3bc` yields output is 2: `A c b 3 b c A` is palindrome

Types of Programming Problems

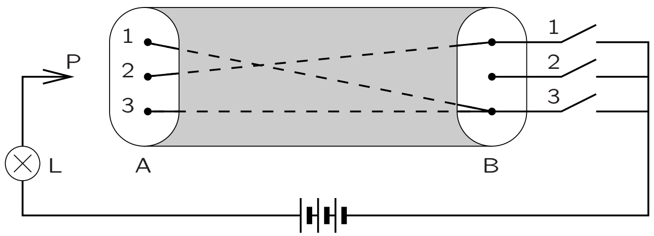
Batch I/O



Reactive I/O



IOI 1995: Wires and Switches



IOI 2000: Median

Given is an odd number of objects, all of distinct strength.

Function *Med3* returns the object of median (middle) strength among three distinct objects:

$$\{a, b, c\} = \{\min\{a, b, c\}, \mathit{Med3}(a, b, c), \max\{a, b, c\}\}$$

Design an efficient algorithm to determine

> the object of median strength among all given objects,

using only function *Med3*.

Engineering Constraints

- **Bounds on input values**

  N = Number of given objects: $5 \leq N \leq 1499$

- **Bounds on computational resources**

  C = Number of calls to *Med3*: $C \leq 7777$

- **Interface to reactive environment**

  Library routines: *GetN* $\quad$ *Med3*(a, b, c) $\quad$ *Answer*(x)

  Robust, secure, experimental mode, trace of dialogue

Definitions

- Set L of object(label)s: $L = \{1, \ldots, N\}$
- **Strength mapping** s from objects to object strengths
- **Weaker than relation** $<_s$ on objects: $i <_s j \Leftrightarrow s(i) < s(j)$
- Objects $\mathsf{min}_s V$, $\mathsf{max}_s V$ are **weakest**, **strongest** in $V \neq \emptyset$
- Object $\mathsf{med}_s V$ has **median strength** in odd-sized V:

$$\#\{ i \in V \mid i <_s \mathsf{med}_s V \} \quad = \quad \#\{ i \in V \mid \mathsf{med}_s V <_s i \}$$

Observations

- Desired object of median strength is uniquely determined

- Consider odd-sized V and $W \subseteq V$ with $\#W \geq 3$

  A median of three-or-more is not an extreme:

$$\mathsf{min}_s V <_s \mathsf{med}_s W <_s \mathsf{max}_s V$$

  The median is invariant under elimination of the extremes:

$$\mathsf{med}_s V \quad = \quad \mathsf{med}_s(V - \{ \mathsf{min}_s V, \mathsf{max}_s V \})$$

- The median of a singleton is easy: $\mathsf{med}_s\{a\} = a$

Onion Peeling

const $L = \{1, \ldots, GetN\}$; **var** V, W: **set of** int

$V := L$
invariant $odd(\#V) \wedge \mathsf{med}_s V = \mathsf{med}_s L$; **variant function** $\#V$
while $\#V \neq 1$ **do** assert $\#V \geq 3$
    $W := V$
    invariant $\{ \mathsf{min}_s V, \mathsf{max}_s V \} \subseteq W \subseteq V$; **variant function** $\#W$
    while $\#W \neq 2$ **do** assert $\#W \geq 3$
        choose $a, b, c \in W$, all distinct
        $W := W - \{ Med3(a, b, c) \}$
    assert: $W = \{ \mathsf{min}_s V, \mathsf{max}_s V \}$
    $V := V - W$
assert: $V = \{ \mathsf{med}_s L \}$
Answer (the object in V)

Total Ordering

- Ordering = Sorting modulo up/down
- Selection sort
- Insertion sort
- Merge sort
- Quicksort
- Heap sort

Insertion Ordering

- Insert x into ordered list of k objects
- $\mathit{Med3}(a, x, b)$ with a, b among ordered k
- Choosing a, b:
  - both at right end: linear search
  - both near middle; binary search
  - at one third and two third: ternary search

Improved Insertion Approaches

Consider situation where $(N + 1)/2$ objects have been inserted.

After inserting another object, the list can be reduced:

- **Half List**
- **Zoom List**

Number of $\mathit{Med3}$-calls for Insertion Ordering on $N = 1499$

Insertion Method	List Variant	Worst case	Average case
Linear	Full	561749	282532
	Half	421499	169655
	Zoom	281623	141676
Binary	Full	12953	11680
	Half	12477	11492
	Zoom	11481	10471
Ternary	Full	9399	8977
	Half	9399	8522
	Zoom	8319	8041

Linear algorithms

- **Expected-Time Linear**: Hoare's Find with two pivots

  Various pivot choices: Straddled, First, Proportional, Random

- **Worst-Case Linear**: Blum, . . . , Tarjan (1972)

  Large constant

Heap-Based Algorithms

A set of objects is called a heap if it supports these operations:

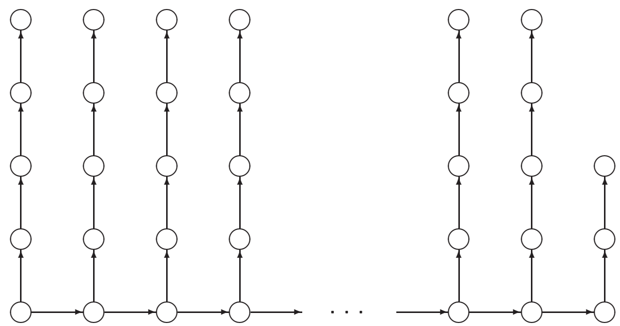
- $BuildHeap(V)$: returns a heap for the set V
- $ReplaceMin(H, x)$:
  - returns H if $x <_s \min_s H$,
  - returns 'H in which $\min_s H$ is replaced by x' if $x >_s \min_s H$;
  - the size of the heap remains the same
- $GetMin(H)$: returns $\min_s H$

Generic Half-Heap Algorithms

var H: **heap of** int
  N, m, x: int

$N := GetN$
$m := (N + 1)$ **div** 2
$H := BuildHeap$ ($\{ 1, \ldots, m \}$)
assert: $\#\{ 1 \le i \le m : \min_s H <_s i \} = m - 1$
for $x := m + 1$ **to** N **do**
  $H := ReplaceMin(H, x)$
  invariant $\#\{ 1 \le i \le x : \min_s H <_s i \} = m - 1$
end
assert: $\#\{ 1 \le i \le N : \min_s H <_s i \} = m - 1 = \#\{ 1 \le i \le N : \min_s H >_s i \}$
$Answer$ ($GetMin(H)$)

Two-Dimensional Ordered List



Worst-case number of $Med3$-calls for $N = 1499$ is at most 6816.

Performance of Algorithms on Passed Test Cases

Case #	1	2	3	4	5	6	7	8	9	10	
N	5	177	577	975	1087	1267	1357	1415	1415	1499	
Alg	M	R	N	R	R	R	R	R	A	R	**Score**
OPE	4	7744									20
LISF	4	4062	619								30
LISH	3	2590	598								30
LISZ	3	2160	598								30
BISF	4	861	4175	7051							40
BISH	5	843	4108	6803							40
BISZ	4	730	3621	6269	7078						50
TISF	3	712	2918	5415	6143	7376					60
TISH	3	669	2707	5349	6011	7103	7642				70
TISZ	3	609	2537	4889	5540	6641	7191	7511	7572		90
TPFS	3	517		1525	2842	3257	3531	2231		3218	80
TPFF	4	395		2205	2378	3635	3601	2663		2493	80
TPFP	5	331	848	3512	1705	2291	3093	2860	2863	2985	100
TPFR	4	372	1778	2201	2507	2981	3377	3987	3279	3540	100
2LHH	4	491	1954	3242	3605	4258	4578	4824	4149	5147	100
QLHH	4	508	2218	3184	3902	4517	4862	5074	4389	5354	100

Invitation

- Informatics is not a compulsary exam topic; should it be?
- Solving algorithmic programming problems is a big challenge
- Designing such problems is an even bigger challenge
- The IOI needs good problems to advertise informatics
- Your ideas are welcome