Sorting Networks

A Lecture in CE Freshman Seminar Series:
Ten Puzzling Problems in Computer Engineering
About This Presentation

This presentation belongs to the lecture series entitled “Ten Puzzling Problems in Computer Engineering,” devised for a ten-week, one-unit, freshman seminar course by Behrooz Parhami, Professor of Computer Engineering at University of California, Santa Barbara. The material can be used freely in teaching and other educational settings. Unauthorized uses, including any use for financial gain, are prohibited. © Behrooz Parhami

<table>
<thead>
<tr>
<th>Edition</th>
<th>Released</th>
<th>Revised</th>
<th>Revised</th>
<th>Revised</th>
<th>Revised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>May 2012</td>
<td>May 2015</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Railroad Tracks and Switches
Coupling and Decoupling of Train Cars

Train cars and engines can be coupled and decoupled quickly.

An engine can push a string of cars, or pull a desired subset by decoupling them from the rest.
Railroad Yards Have Many Tracks and Switches
Rearranging Trains

Sorting algorithm: Assemble train in stub, beginning with the last car
Repeat: If the next car is X, decouple train after X, push X into stub
Model Railroad Yard
Rearrangement with Change of Direction

What types of sorting are possible with a turnaround loop? A wye?

Configuration after the first few steps in sorting the train

Complete the sorting process, assuming that it is okay for the engine to end up behind the train
Model Train Turntable
Delivering Train Cars in a Specific Order

Cars in the train below have been sorted according to their delivery points. However, it is still nontrivial to deposit car A in stub 1, car B in stub 2, and car C in siding 3. Cars can be pulled or pushed by the engine.

Is there a better initial ordering of the cars for the deliveries in this puzzle?
Train Passing Puzzle

The trains below must pass each other using a siding that can hold only one car or one engine. Show how this can be done.
Fast Combining or Reordering of Train Cars

Forming multiple trains from incoming cars

Train reordering via a set of stubs

Cars are pushed or pulled by an engine

Alternatively, movement in one direction may be achieved via sloped rails

Switches used to be adjusted manually, but nowadays, electronic control is used
Sorting Train Cars in Parallel

Track 1
Track 2
Track 3
Track 4

Target track
Unconditional exchange
Compare-exchange

Is adding this compare-exchange element sufficient for producing a valid sorting network?
Validating a Sorting Network

In the example above, it was fairly easy to show the validity of the sorting network. Generally, it is much more difficult.

How would one establish the validity of this 16-input sorting network?

More importantly, how does one come up with this design in the first place?
The Zero-One Principle

A sorting net built of comparators is valid if it correctly sorts all 0-1 sequences

So, we can validate a sorting network using $2^n$ rather than $n!$ input patterns

$n = 12$: $2^n = 4096$, $n! = 479,001,600$ (thousands vs. half a billion)
A 16-Input Sorting Network

Use 4-input sorters, follow by (4, 4)-mergers, and end with an (8, 8)-merger

Using the 0-1 principle, we can validate this network via 16 + 25 + 81 tests

4-sorter tests  (4, 4)-merger tests  (8, 8)-merger tests
Insertion Sort and Selection Sort

Fig. 7.8 Sorting network based on insertion sort or selection sort.

<table>
<thead>
<tr>
<th>C(n) = n(n - 1)/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>D(n) = 2n - 3</td>
</tr>
<tr>
<td>Cost × Delay = Θ(n^3)</td>
</tr>
</tbody>
</table>

May 2015

Sorting Networks

B. Parhami

UCSB
The Best Sorting Networks

Criterion 1: The number of sticks or compare-exchange blocks (cost)
Criterion 2: The number of compare-exchanges in sequence (delay)
Criterion 3: The product of cost and delay (cost-effectiveness)

The most cost-effective \( n \)-input sorting network may be neither the fastest design, nor the lowest-cost design
Electronic Sorting Networks

Electronic sorting networks are built of 2-sorters, building blocks that can sort two inputs.

Applications of sorting networks:
Directing information packets to their destinations in a network router
Connecting \( n \) processors to \( n \) memory modules in a parallel computer.