On the Cost-Optimal Parallel Solution of the Majority Problem

Jie Wu
Center for Networked Computing
Temple University, USA

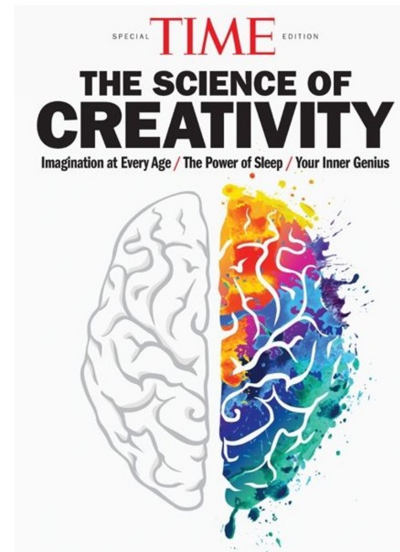
2023 IEEE APDCM Keynote

Outline

1. On Creativity
2. Majority Problem
3. Existing Solutions
4. Elimination Solution
5. Parallel Elimination
6. Conclusions and Open Problems

1. On Creativity

- Beyond ChatGPT
  - Machine-generated content
  - Cannot replace creative research
- On creativity
  - Out-of-the-box thinking
    - The science of creativity [1]
    - Cultures of creativity [2]



[1] Time, Special Issue on The Science of Creativity

[2] Alexander V. Humboldt Foundation, Cultures of Creativity

On Utility

Utility of research

- Solve practical issues in society
- Overemphasis on utility can be the enemy of creativity

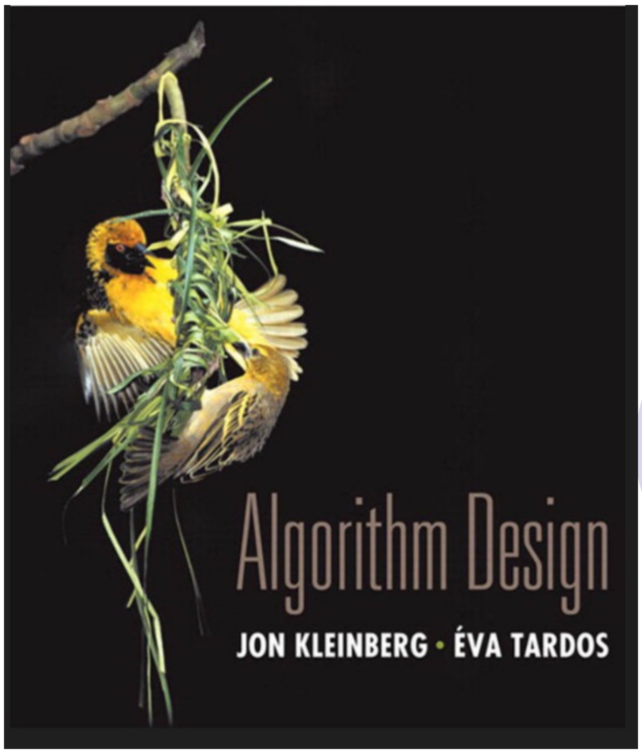
Balance of creativity and utility

- Be curious, relentlessly curious
- Indulge fantasy, let your reach exceed your grasp
- Being creative can make life more enjoyable

2. Majority Problem

- There are n bank cards that belong to different bank accounts.
- A tester checks two cards and returns
  - "yes" if they match, i.e., belong to the same bank account
  - "no" if they mismatch
- Use a minimum number of tests to determine if > n/2 cards belong to the same bank account and then find them.

Textbook Exercise



3. Suppose you're consulting for a bank that's concerned about fraud detection, and they come to you with the following problem. They have a collection of n bank cards that they've confiscated, suspecting them of being used in fraud. Each bank card is a small plastic object, containing a magnetic stripe with some encrypted data, and it corresponds to a unique account in the bank. Each account can have many bank cards corresponding to it, and we'll say that two bank cards are *equivalent* if they correspond to the same account.

It's very difficult to read the account number off a bank card directly, but the bank has a high-tech "equivalence tester" that takes two bank cards and, after performing some computations, determines whether they are equivalent.

Their question is the following: among the collection of n cards, is there a set of more than $n/2$ of them that are all equivalent to one another? Assume that the only feasible operations you can do with the cards are to pick two of them and plug them in to the equivalence tester. Show how to decide the answer to their question with only $O(n \log n)$ invocations of the equivalence tester.

US House Speaker Election

- Voting and counting is a sequential process !

2023 Speaker of the United States House of Representatives election

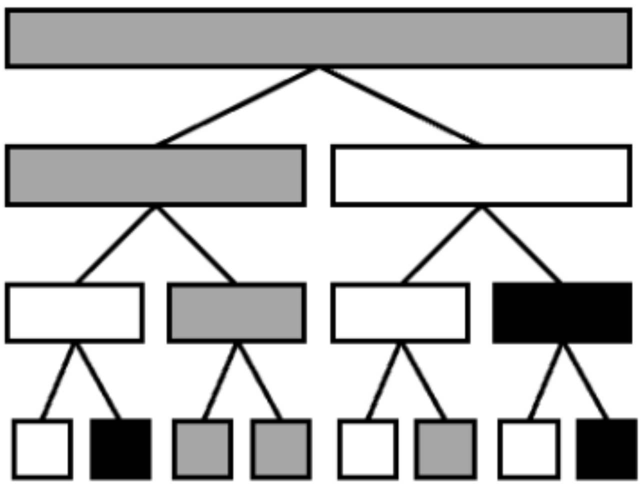
← 2021 | **January 3–7, 2023**

Needed to win: Majority of votes cast
First ballot: 434 votes cast, 218 needed for a majority
Fifteenth ballot: 428 votes cast, 215 needed for a majority

Candidate	**Kevin McCarthy**	Hakeem Jeffries
Party	Republican	Democratic
Leader's seat	California 20th	New York 8th
First ballot	203 (46.8%)	212 (48.8%)
Fifteenth ballot	**216 (50.5%)**	212 (49.5%)

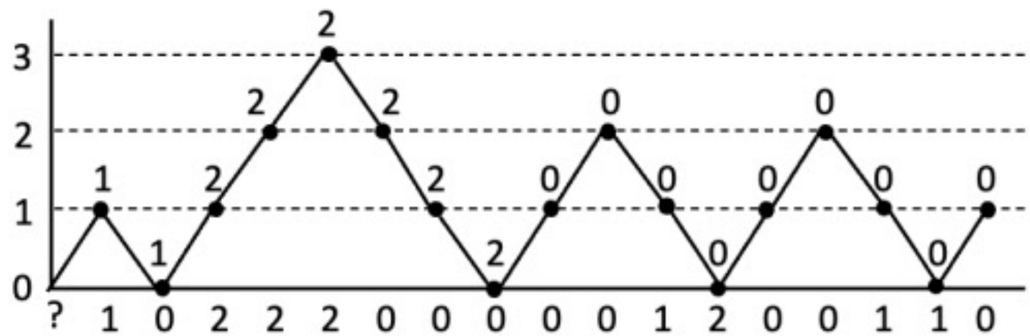
3. Existing Solutions

- Divide-and-Conquer Solution
  - O(n log n): log n levels, n operations per level
  - E.g., white: no majority, black: majority, gray: majority



Boyer-Moore Solution

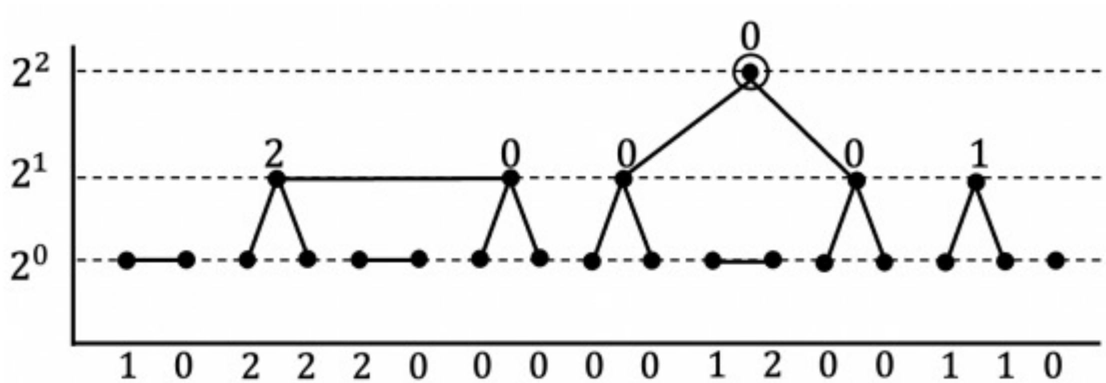
- Linear solution O(n): one storage unit with a counter
  - Phase 1: candidate (also seed) selection
  - Phase 2: candidate validation (can be a false positive)



  - Seed 0 in 10222000012001100 is a majority
  - Seed 0 in 1212122200 is not a majority (false positive)

4. Elimination Solution

- Cards are pair-wise compared
  - *Promoted* (to a higher level): if they match
  - *Eliminated*: if they mismatch
  - *Survived*: single element left at a level



Theorem 1: If the given set of cards has a majority account, that account must be the top-level survivor after elimination.

Theorem 2: The elimination solution is linear, i.e., O(n).

4. Parallel Elimination

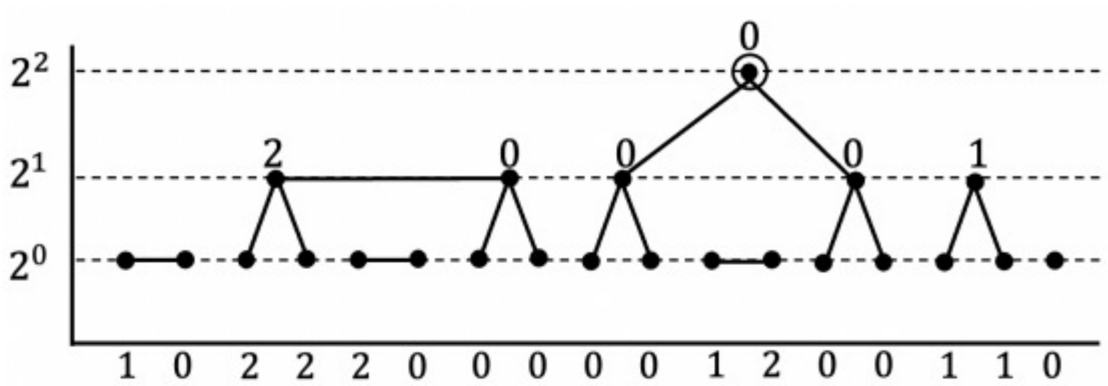
- Basic metrics
  - Cost: product of its run time and the number of processors
  - Cost-optimal: cost matches the run time of the best-known sequential solution

- Cost-optimal parallel solution
  - Cost of Boyer-Moore: O(n), but it is slow
  - Speed of Divide-and-conquer: O(log n), but cost more O(n log n)

- Our goal of cost: O(n), with time: O(log n)
  - using p: O(n/log n) processors (i.e., testers)

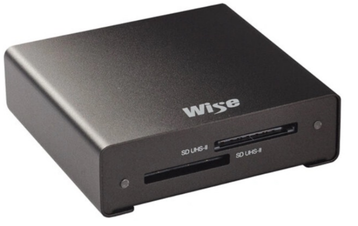
Challenges

- Keeping all processors busy
- A solution with seed copying
  - A tester can copy a seed one at a time, total O(log p) rounds
- Load balance
  - Each tester handles O(log n) elements
  - Balance at both elimination and validation phases



Without Seed Copying

- How to quickly find
  - p cards that match the seed?

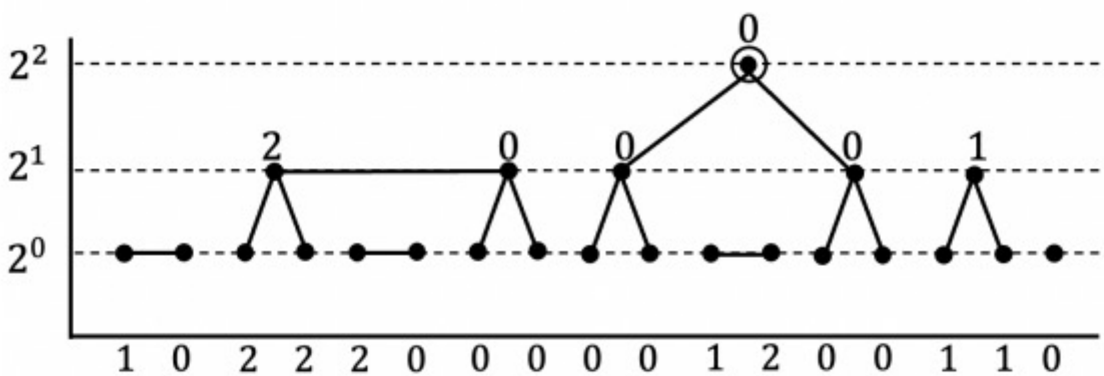


- A mismatch
  - between a seed with a new card does not create a new seed!
- Solution
  - Card re-sequencing to increase the match probability

Card Re-Sequencing

(a) Rearrange the sequence: the top level down to the lowest level, and within a level, going from left to right.

(b) Move the first non-seed element to the end.

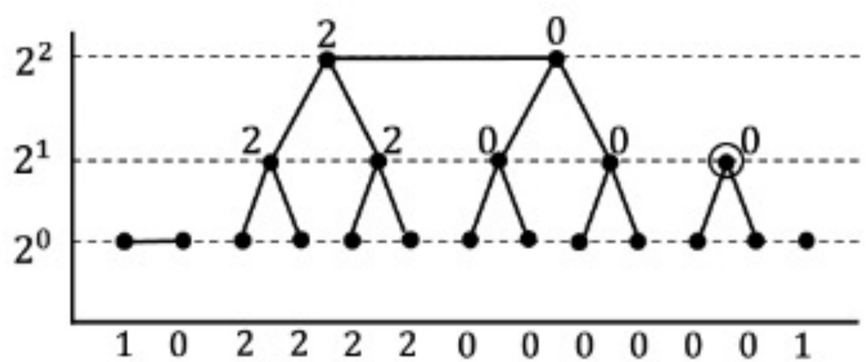


(a) Re-sequencing step (a)

(b) Re-sequencing step (b)

Card Re-Sequencing (cont'd)

Theorem 3: In the new card sequence, the majority has at least half in any prefix subsequence.



Step (a): 2222000000101

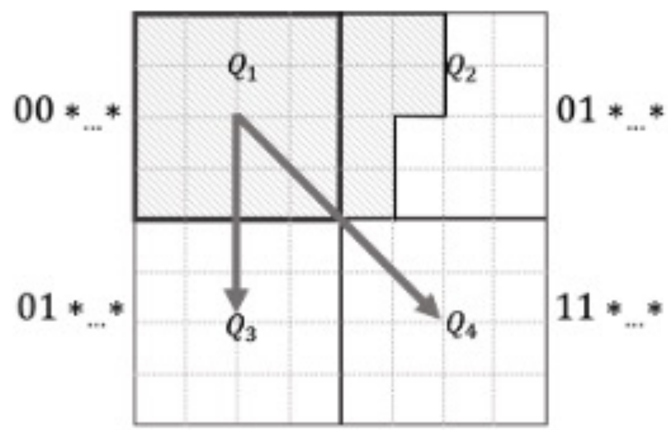
Step (b): 0000001012222

Parallel Exploration

Double the area each round

Round 1: seed 00 (i.e., 0) checks cards 01, 10, and 11.

Round 2: seed 000 checks cards 100 and 110; seed 001 checks 101 and 111.



...

Round i: For a seed from 0 to $2^{i-1} - 1$, it is an (i + 1)-bit binary 00 *...*, where *, 0 or 1, is a wild card. This seed checks cards 01*...* and 11*...*.

5. Conclusions and Open Problems

- Majority Problem
  - Proposed elimination solution
  - Cost-efficient parallel solution: card re-sequencing
- Open Problems
  - Cost-optimal parallel solutions without an atomic counter
  - Cost-optimal parallel solution in a strong sense, i.e., fastest

Questions