COT 6936: Topics in Algorithms

Giri Narasimhan

ECS 254A / EC 2474; Phone x3748; Email: giri@cs.fiu.edu
HOMEPAGE: http://www.cs.fiu.edu/~giri
http://www.cs.fiu.edu/~giri/teach/COT6936_S14.html
https://moodle.cis.fiu.edu/v2.1/course/view.php?id=612

January 7, 2014

Presentation Outline

Purpose of this course

First course in Algorithms is inadequate for PhD students ...

Purpose of this course

First course in Algorithms is inadequate for PhD students ...

- Need to go beyond standard techniques and problems
- Need to go beyond basic analysis techniques

Purpose of this course

First course in Algorithms is inadequate for PhD students ...

- Need to go beyond standard techniques and problems
- Need to go beyond basic analysis techniques

So, we will ...

Purpose of this course

First course in Algorithms is inadequate for PhD students ...

- Need to go beyond standard techniques and problems
- Need to go beyond basic analysis techniques

So, we will ...

- Model/formalize problems
- Look for existing solutions or create new ones

Expectations

- Attend class
- Participate in class
- Team work and discussion groups
- Solve practical research problems
- Make a presentation; write a report
- Write a research paper
- No cell phones, SMS or email during class

Evaluations

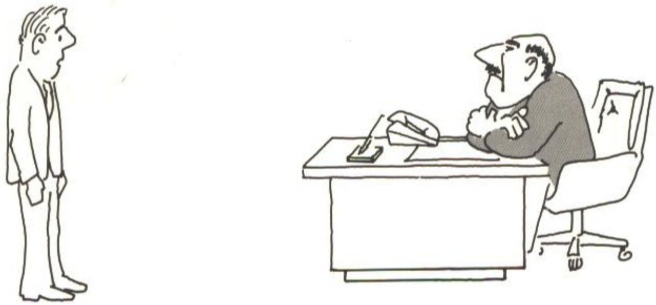
20%	Exam
5%	Quizzes
15%	Homework Assignments
40%	Semester Project
20%	Class Participation

Semester Milestones

by Jan 21	Discuss a project
by Jan 28	Email me on details of selected project and team
Feb 25	Short presentation on selected project
Mar 21-23	Take-home Exam
Apr 8-10	Final Project Presentations
by Apr 17	Submit Final Project Report

Presentation Outline

Why care about Algorithms?

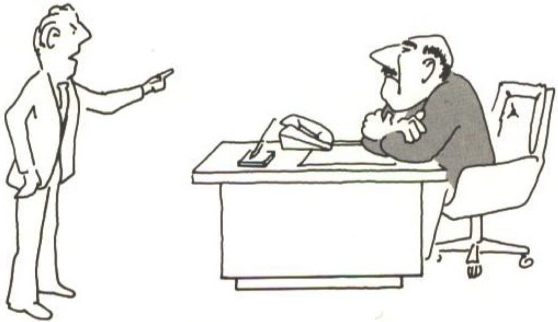


“I can’t find an efficient algorithm, I guess I’m just too dumb.”

Cart

from Garey and Johnson’s “Guide to Intractability”.

Why care about Algorithms?



"I can't find an efficient algorithm, because no such algorithm is possible!"

Ca

from Garey and Johnson's "Guide to Intractability".

Why care about Algorithms?

"I can't find an efficient algorithm, but neither can all these famous people."

Cartoon from Garey and Johnson's "Guide to Intractability".

Presentation Outline

Classical Algorithms Model

- Input-Output description provided
- Input provided and stored in memory
- Output computed and stored/output
- Program stored in memory
- Algebraic computation-tree model
- Sequential and deterministic algorithms
- Worst-case time and space analysis

Classical Algorithms Model

- Input-Output description provided
- Input provided and stored in memory
- Output computed and stored/output
- Program stored in memory
- Algebraic computation-tree model
- Sequential and deterministic algorithms
- Worst-case time and space analysis
- But this is very limiting ...

Newer Algorithms Model

- Input stored in memory
- Sequential algorithms
- Deterministic algorithms
- Worst-case analysis

Newer Algorithms Model

- Input stored in memory
- Sequential algorithms
- Deterministic algorithms
- Worst-case analysis

Newer Algorithms Model

- Input stored in memory
- Sequential algorithms
- Deterministic algorithms
- Worst-case analysis

- On-line / Streaming Algorithms

Newer Algorithms Model

- Input stored in memory
- Sequential algorithms
- Deterministic algorithms
- Worst-case analysis

- On-line / Streaming Algorithms
- Parallel / Randomized algorithms

Newer Algorithms Model

- Input stored in memory
- Sequential algorithms
- Deterministic algorithms
- Worst-case analysis

- On-line / Streaming Algorithms
- Parallel / Randomized algorithms
- Amortized / Randomized Analysis

Newer Algorithms Model

- Input stored in memory
- Sequential algorithms
- Deterministic algorithms
- Worst-case analysis

- On-line / Streaming Algorithms
- Parallel / Randomized algorithms
- Amortized / Randomized Analysis
- Limited resource algorithms; External memory algorithms; cache-aware algorithms, ...

On-line Algorithms

- Input revealed over time
- Next decision must be based on input seen so far
- Realistic problem setting

Streaming Algorithms

- It's an on-line algorithm
- Input too much and too fast
- Cannot store entire input after reading.
- Cannot process entire input for each query

MAX-GAP Problem

- What does the floor/ceiling functions operation buy us.
- Demonstrable speed up with the floor operation.

Parallel Algorithms

- Come in different flavors: SIMD, MIMD, Distributed
- Concept of speedup

Randomized Algorithms

- Some steps are ramdomized
- Requires average-case analysis

Randomized Algorithms

- Does it pay off?
- Quick sort

Amortized Analysis

- Worst-case cost averaged over a series of steps

Randomized Analysis

- A ship arrives at a port. 40 sailors go ashore for revelry. They return to the ship rather inebriated. Being unable to remember their cabin location, they find a random unoccupied cabin to sleep the night. How many sailors are expected to sleep in their own cabins?
- Variants? Generalizations?

Presentation Outline

Additional Topics

- Randomized Algorithms
- Online Algorithms (Computing Systems)
- Computational Geometry
- Approximation Algorithms
- Computational Biology / Finance
- Combinatorial Optimization
- Algorithmic Game theory
- ...

Presentation Outline

LEDA: Library of Efficient Algorithms and Data Structures

- Free edition available from: `http://www.algorithmic-solutions.com/leda/index.htm`

LEDA: Library of Efficient Algorithms and Data Structures

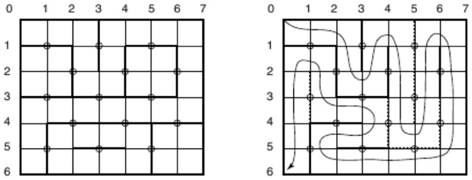
- Free edition available from: `http://www.algorithmic-solutions.com/leda/index.htm`
- Professional/Research Editions cost 1200 Euros.

Presentation Outline

Fix The Pond

12529 Fix the Pond

A shrimp farm uses a rectangular pond built as a grid with $2N$ rows and $2N+1$ columns of square cells, for a given integer N. Each cell side is one meter long. The pond has exactly $(2N-1) \times N$ barriers of length two meters, used to temporarily isolate smaller sections inside the pond for breeding different kinds of shrimp. The barriers have their middle points fixed precisely at the integer coordinates (a, b), for all $0 < a < 2N$ and $0 < b < 2N+1$, where both a and b are odd, or both are even. Each barrier can be rotated around its middle point to change the pond configuration; however, by being rotated, a barrier switches between only two possible positions, always parallel to the pond sides, vertical or horizontal. The left part of the figure below shows a pond configuration, with $N = 3$.



At the end of every season the pond is closed for maintenance and cleaning. It must then be reconfigured so that a special machine can sweep the pond floor. The machine starts its work at the top left cell, and needs to pass through every cell exactly once, finishing in the bottom left cell. The right part of the figure shows one such reconfiguration, where six barriers were switched. For this example, though, four barrier switches would have been enough.

You must write a program that given a pond configuration, determines the minimum number of barrier switches needed to reconfigure the pond as specified above. There is always at least one possible way to reconfigure the pond as specified.

Fix The Pond

Input

Each test case is described using several lines. The first line contains an integer N indicating that the pond has $2N$ rows and $2N+1$ columns ($1 \leq N \leq 300$). Each of the next $2N-1$ lines contains a string of N characters describing the orientation of the barriers. In the i-th line, the j-th character indicates the orientation of the barrier whose middle point is at coordinates $(i, 2j-1)$ if i is odd, or $(i, 2j)$ if i is even, for $i = 1, 2, \ldots, 2N-1$ and $j = 1, 2, \ldots, N$. The character is the uppercase letter 'V' if the orientation is vertical, or the uppercase letter 'H' if it is horizontal.

Output

For each test case output a line with an integer representing the minimum number of barrier switches needed to reconfigure the pond as specified.

Sample Input

```
3
HVH
VVV
HHH
HHH
VHV
1
H
1
V
```

Sample Output

```
4
0
1
```

More Challenge Problems

- Robot Challenge Problem: `http://users.cis.fiu.edu/~giri/teach/6936/S10/SER2009_RobotChallenge.pdf`
- Frequency Count Problem: `http://cs05.informatik.uni-ulm.de/acm/Locals/2007/html/frequent.html`
- Profits: `http://www.cs.fiu.edu/~giri/teach/6936/S12/SER2010Profits.pdf`
- Family Fortune (see Problem H. from the problem set): `http://www.cs.fiu.edu/~giri/teach/6936/S12/SER2011AllProblems.pdf`