Algorithms for Programming Contests SS20 - Week 06

Mikhail Raskin, Tobias Meggendorfer, Stefan Jaax,
A.R.Balasubramanian, Christoph Welzel

June 2, 2020

A: Alarm Clock - Sample Solution

Problem

Find all compatible starting times for some given time sequence of a broken alarm clock.

A: Alarm Clock - Sample Solution

- There are at most $24 \cdot 60$ possible times, brute force is possible.
- Check all possible start times, increase by 1 minute for every observation
- Every segment that lit up at least once is *working*
- Every segment that should light up but didn't is *broken*
- For every simulation, keep a list of broken and working segments.
- If a segment is both *broken* and *working* in the same simulation, that start time is not possible.

B: Muffin Queen - Sample Solution

Problem

All bakers can choose either recipe 1 or 2.
You want to satisfy all judges.
All judges want to have at least one of their preferences.

B: Muffin Queen - Sample Solution

Problem

All bakers can choose either recipe 1 or 2.(Boolean Variables)
You want to satisfy all judges.(Conjunction)
All judges want to have at least one of their preferences.(Disjunction)

Called "Boolean Satisfiability Problem" and is **NP**-hard

B: Muffin Queen - Sample Solution

- SAT problem in CNF with m variables and n clauses
- True/false corresponds to first/second recipe
- At most 20 variables and 100 clauses
- $2^m = 2^{20} \approx 10^6$ possible assignments
- Testing one assignment takes at most $m \cdot n = 2000$ operations
- Worst case input with 20 testcases:
  $20 \cdot 20 \cdot 100 \cdot 2^{20} \approx 4 \cdot 10^{10}$ operations
- Can be slightly too slow for naive simulation using brute force
  (1 GHz $= 10^9$ Hz (operations per second))
- Feasible using backtracking with early detection of both satsfiability and unsatisfiability (e.g. DPLL)

B: Muffin Queen - Competitions

- Input format is used in SAT competitions (one test case per file, no leading 1)
- Test cases are much bigger there

C: Planetarium Problem - Sample Solution

Problem

Find all minimal subdivisions of a large number of stars.
Subdivisions are prime.

C: Planetarium Problem - Sample Solution

- Find prime factors of a (big) integer.
- Maximum value 10^{85}, 282 bits.
- Brute force algorithm checks all divisors up to $\sqrt{n}$.
- Complexity UP $\cap$ coUP.
- Encryption algorithms use that factoring of big numbers is hard.
- Most PGP keys have lengths between 512 and 2048 bits.
- **Triple DES provides 112 bits of security**, but is *still* considered to be secure.
- Not solvable in short time.

D: Queens Problem - Sample Solution

Problem

Given an initial configuration of a chessboard, find a solution to the n-Queens problem.

D: Queens Problem - Sample Solution

- Problem is **NP**-hard[1]
- For $n = 15$, there are only 2 279 184 valid assignments.
- Iterate over rows, then columns (or vice versa).
- Check for queens in same row/column/diagonals.
- Return if all rows (columns) are processed or if for one row no queen could be set.
- Take care of already set queens.

[1] Gent *et al.: Complexity of n-Queens Completion (2017)*

E: Story Time - Sample Solution

Problem

Given a partial order of chapters, find the number of possible total orderings.
(corresponds to, i.e., Thread Scheduling)

Solution

- Problem is #**P**-hard[2]
- For 13 chapters there are at most 13! orderings, which is $\approx 6 \times 10^9$
- For each character $c_{i,1} < c_{i,2} < \ldots$
- Since there are at most 6 characters, that creates at least 7 constraints.
- $\Rightarrow$ tractable by backtracking

[2] Brightwell and Winkler: *Counting Linear Extensions* (1991)

E: Story Time - Sample Solution

Problem

Given a partial order of chapters, find the number of possible total orderings.
(corresponds to, i.e., Thread Scheduling)

Solution

- Problem is **#P**-hard[2]
- For 13 chapters there are at most 13! orderings, which is $\approx 6 \times 10^9$
- For each character $c_{i,1} < c_{i,2} < \ldots$
- Since there are at most 6 characters, that creates at least 7 constraints.
- $\Rightarrow$ tractable by backtracking

[2] Brightwell and Winkler: *Counting Linear Extensions* (1991)

E: Story Time - Sample Solution

Approach

- For a state of written chapters, find which chapters are enabled
  - All predecessors already written
  - Last written chapter for different character
- Try out all enabled chapters

Careful

Do NOT copy the partial solutions, but extend them.
(C++: always use reference parameters, Java: don't *Arrays.copy* the partial solution)

F: Hockey Champion - Sample Solution

Problem

In a graph, find a maximal subgraph where every node is connected to each other.

Called "Maximum Clique Problem" and is **NP-hard**.

F: Hockey Champion - Sample Solution

- Graph has n nodes and m edges.
- At most 1000 nodes and 100000 edges.
- Clique of size k has $\frac{k(k-1)}{2}$ edges.
- Clique size of up to 447 possible.
- Up to $\sum_{k=0}^{447} \binom{1000}{k} \approx 4.8 \cdot 10^{296}$ possible cliques to check.
- Currently, the best known algorithm for the Maximum Clique Problem runs in $O(1.1888^n)$ time.
- Too big to solve in short time.

G: Trapped - Sample Solution

Problem

Find a simple path in the given cave collecting all tools.

G: Trapped - Sample Solution

- Problem is **NP**-hard (reduction from Hamilton path).
- 25 walkable fields, 4 directions for each field: 4^{25} paths.
- Better approximation: At most 10 fields will have 3 free neighbours, all others at most 2: $3^{10} \cdot 2^{15} \approx 2 \cdot 10^9$ paths.
- Check all paths by backtracking:
  - Check the four neighbouring squares.
  - Move to each square if possible (walkable and not used yet).
  - Return if there is a way such that all tools are collected.