ACM International Collegiate Programming Contest — Training Session III

C. Maria Keet

Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

August 27, 2016

Today

- Graphs (other than just plain Dijkstra)
- Some more and less common data structures
- Practice with the Arab Regional 2013

Outline

As long as I learn, I live

- UVa 459 Graph connectivity / 12376 As long as I learn, I live
- Graphs

As long as I learn, I live

- UVa 459 Graph connectivity / 12376 As long as I learn, I live
- Graphs

$\Rightarrow$ Finding connected components type of problem, using DFS or BFS

$\Rightarrow$ See also §4.2.3 of CP3

Designing Railway routes

- 2013 regionals problem (see printout), few solved it
- type: algorithmically, somewhat challenging
- Solve it

Designing Railway routes

- 2013 regionals problem (see printout), few solved it
- type: algorithmically, somewhat challenging
- Solve it

$\Rightarrow$ My direction of solution is one of 'modify existing algorithm'

Designing Railway routes: toward solution

- Task: second-best solution
- Has lots of constraints on the input data
- Check first example, and draw and check the second example (in the input/output files, given)

Designing Railway routes: toward solution

- Task: second-best solution
- Has lots of constraints on the input data
- Check first example, and draw and check the second example (in the input/output files, given)
- Do you have to start at node 1?

Designing Railway routes: toward solution

- Task: second-best solution
- Has lots of constraints on the input data
- Check first example, and draw and check the second example (in the input/output files, given)
- Do you have to start at node 1? no.
- What algorithms do we have to compute trees and shortest paths? Can we use that here?

Designing Railway routes: toward solution

- Task: second-best solution
- Has lots of constraints on the input data
- Check first example, and draw and check the second example (in the input/output files, given)
- Do you have to start at node 1? no.
- What algorithms do we have to compute trees and shortest paths? Can we use that here?
- OSPF/Dijkstra's algorithm, spanning tree algorithms

Designing Railway routes: toward solution

- Task: second-best solution
- Has lots of constraints on the input data
- Check first example, and draw and check the second example (in the input/output files, given)
- Do you have to start at node 1? no.
- What algorithms do we have to compute trees and shortest paths? Can we use that here?
- OSPF/Dijkstra's algorithm, spanning tree algorithms
- But can we get the 2nd-best from that?
  - Compute OSPF for each node, take the 2nd-lowest value, or mutate lowest value?
  - Modify a greedy algorithm to find 'second-best'?
  - Which one works best? or neither, and use another solution?

Another step toward solution

- Dijkstra (or other link state algorithm): no (why not?)
- Bellman-Ford (distance vector algorithm)? no (same reason as previous)

Another step toward solution

- Dijkstra (or other link state algorithm): no (why not?)
- Bellman-Ford (distance vector algorithm)? no (same reason as previous)
- need a **minimum spanning tree**, and take the 2nd best option
- Or perhaps compute all, rank them on cost, take the 2nd best?

Another step toward solution

- Dijkstra (or other link state algorithm): no (why not?)
- Bellman-Ford (distance vector algorithm)? no (same reason as previous)
- need a **minimum spanning tree**, and take the 2nd best option
- Or perhaps compute all, rank them on cost, take the 2nd best?
- What is a minimum spanning tree?

- Which algorithms do we have?

Another step toward solution

- Dijkstra (or other link state algorithm): no (why not?)
- Bellman-Ford (distance vector algorithm)? no (same reason as previous)
- need a **minimum spanning tree**, and take the 2nd best option
- Or perhaps compute all, rank them on cost, take the 2nd best?
- What is a minimum spanning tree?
  - A spanning tree of a connected, undirected, graph is a subgraph that is a tree and connects all the vertices together.
- Which algorithms do we have?

Another step toward solution

- Dijkstra (or other link state algorithm): no (why not?)
- Bellman-Ford (distance vector algorithm)? no (same reason as previous)
- need a **minimum spanning tree**, and take the 2nd best option
- Or perhaps compute all, rank them on cost, take the 2nd best?
- What is a minimum spanning tree?
  - A spanning tree of a connected, undirected, graph is a subgraph that is a tree and connects all the vertices together.
- Which algorithms do we have?
  - Skiena (pp192-202): Prim's algorithm (also greedy), Kruskal's (also greedy, but doesn't start with a particular vertex)
  - Longer list: `https://en.wikipedia.org/wiki/Minimum_spanning_tree#Algorithms`

Solution (sketch)

1. Kruskal for MST
   `https://www.youtube.com/watch?v=71UQH7Pr9kU`
2. For each of the non-MST edges:
   3. Take edge not in MST that results in a cycle, ϵ
   4. Compare weight ϵ with heaviest (non-ϵ) edge in the cycle
   5. If lowest value so far, then store as lowest solution σ
6. add ϵ of best solution σ and remove the heaviest (non-ϵ) edge of that cycle

Software Allocation

- UVa 259 Software Allocation
- Graphs
- Note: this needs a different class of graph algorithms cf. what we have seen so far
- Figure out how your graph look like,
- What is different about it cf. the others?

Software Allocation

- UVa 259 Software Allocation
- Graphs
- Note: this needs a different class of graph algorithms cf. what we have seen so far
- Figure out how your graph look like,
- What is different about it cf. the others?

$\Rightarrow$ You need a 'network flow' algorithm for this

What is a 'network flow'?

- Directed graph, where each edge has a max capacity
- Amount of 'flow' into the node = amount of 'flow' out of the node, unless it's the source or sink
- Used for modelling traffic systems, fluids in pipes, bits across wires...
- Can we indeed model the Software Allocation problem as a network flow one? Show it.
- Then: which algorithm to compute the max flow?

What is a 'network flow'?

- Directed graph, where each edge has a max capacity
- Amount of 'flow' into the node = amount of 'flow' out of the node, unless it's the source or sink
- Used for modelling traffic systems, fluids in pipes, bits across wires...
- Can we indeed model the Software Allocation problem as a network flow one? Show it.
- Then: which algorithm to compute the max flow?
- Edmonds-Karp's algorithm`https://www.youtube.com/watch?v=MczX0SM3I84`
- See also §4.6 of CP3 for a description

Outline

Streaming System

- This is UVa problem 1203, "Argus" (and also used in the Asia - Beijing - 2004/2005 ACM ICPC Local Contest). (ICPC live 6421)
- Main point of this exercise here: Data structures
- Solve it

Streaming System

- This is UVa problem 1203, "Argus" (and also used in the Asia - Beijing - 2004/2005 ACM ICPC Local Contest). (ICPC live 6421)
- Main point of this exercise here: Data structures
- Solve it

$\Rightarrow$ Priority Queue

Young and Successful

- Not an ACM ICPC problem from recent years (or at all)
- Description on printout
- Identify type
- Try to solve it

Young and Successful

- Not an ACM ICPC problem from recent years (or at all)
- Description on printout
- Identify type
- Try to solve it
- Solve it algorithmically

Young and Successful

- Not an ACM ICPC problem from recent years (or at all)
- Description on printout
- Identify type
- Try to solve it
- Solve it algorithmically

$\Rightarrow$ Basically a version of a *range search*:
input: A set S with n points in d-dimensional space E, and a query asking for the points in region Q

Hic-Hac-Hoe

- Slight modification to Tic-Tac-Toe: infinite board size
- data structure & search, with a time limit of 2 sec.
- Solve it

Hic-Hac-Hoe

- Slight modification to Tic-Tac-Toe: infinite board size
- data structure & search, with a time limit of 2 sec.
- Solve it

$\Rightarrow$ instead of a 2D array, use a **balanced BST** to store the o's and x's, and use that BST to check the game state

$\Rightarrow$ Balanced BST amounts to using the Divide & Conquer strategy

- Note: difference with heap is that the tree doesn't have to be complete
- C++ STL: `map` and `set`; Java with `TreeMap` and `TreeSet`

Outline

Looking at a whole contest

- The 2013 ACM-ICPC Arab Regional Programming Contest
- Groups of 3. Consider:
  - Roles: who will do what? Manager/time-keeper?
  - Pair solving and pair programming
  - Categorise problems (recall problem solving strategies)
  - Know your strengths and weaknesses
  - Decide which problem(s) to solve first
- `https://icpcarchive.ecs.baylor.edu/`, "Browse Problems" – "Regionals 2013" – "Africa/Middle East" etc.

Scheduled training dates

- Aug 6: 10:00-16:00
- Aug 13: 10:00-16:00
- Aug 27: 10:00-15:00
- **Sept 10: 10:00-16:00 → Ashraf Moolla on DP**
- Sept 17: 10:00-16:00
- Sept 24: 10:00-16:00 or Oct 1: 10:00-16:00
- Date of the regionals: very likely Oct 15