CS2020 – Data Structures and Algorithms Accelerated

Recitation Week09 – Graph Modeling

stevenhalim@gmail.com

NUS National University of Singapore

School of Computing

Graph Modeling

- Given a problem that may *appear unrelated* to graph
  - Your task is to find the graph!!
    - » Find the **vertices** and **edges** and the **graph problem**!!
  - After that, find the best possible (graph) data structure and algorithm to solve it
    - » Due to lecture scheduling issue, some of the (better) algorithms will only be discussed later, so during this recitation, we will focus on modeling the graph first, and deal with the solution later
- Easier said than done…
  - Let’s see several examples ☺

UVa 429 – Word Transformation (1)

- Given:
  - starting, ending, and list of other words
- Problem: Find **shortest** sequence of 1 character transformation **from starting word** to **ending word**
  - Example:
    - Starting word: **spice**, Ending word: **stock**
    - List of other words (max 200): dip lip mad map maple may pad pip pod pop sap sip slice slick spice stick stock
    - Answer: spice slice slick stick stock (4 transformations)
- No graph in the problem description?

UVa 429 – Word Transformation (2)

- What is the graph?
  - Each **word** is a vertex, index them from [0..up to 199]
  - Connect two vertices (words) with an edge
    if the **Hamming distance** between them is 1
    - i.e. only 1 character difference
  - Store this information in an **Adjacency List** (more efficient)
- What is the graph problem?
  - **SSSP** from source = **starting word**, output **D[ending word]**
  - The graph is unweighted (all edges have weight = 1)
- What is the appropriate graph algorithm?
  - Next Tue, you will know that the answer is O(V + E) BFS

LA 4408 – Unlock the Lock (1)

Kuala Lumpur 2008

- Given:
  - L, current lock code, 4 digits integer 0000-9999
  - U, unlock code, also 4 digits integer 0000-9999
  - R, number of buttons, R <= 10
    - Followed by R_i numbers: the value of each button
    - Each press of button i transform L to $(L+R_i)$ % 10000
- Problem: Can U be reached from L?
  - If yes, what is the **minimum** button presses?
  - If no, print “Permanently Locked”
- No graph in problem description?

LA 4408 – Unlock the Lock (2)

Kuala Lumpur 2008

- Slide 2 of this problem is hidden ☺
  - To be discussed in during the recitation, so don’t skip it ☺

UVa 929 – Number Maze (1)

- Given:
  - An N x M integer maze; each cell is between [0 .. 9]
- Problem: Find a minimum cost to traverse the maze from top left corner to bottom right corner where passing a cell (i, j) incurs a cost of maze[i][j]
  - Example, for this 2 x 3 maze, the answer is in red (cost = 8):

    0 3 6
    7 1 4
- No graph in the problem description?

UVa 929 – Number Maze (2)

- What is the graph (another implicit graph)?
  - Vertex is a pair of (row, col)
  - There are 4 possible edges: (N/E/S/W)
    - Edges have associated weight (the cost of the target cell)
- What is the graph problem?
  - **SSSP** from source = **(0, 0)**, output **D[(M - 1, N - 1)]**
    - Note, initially D[(0, 0)] = maze[0][0]
  - The graph is weighted, non-negative
- What is the appropriate graph algorithm?
  - Next Tue, you will know that the answer is O((V + E) log V) Dijkstra’s

UVa 10603 – Fill (1)

- Given (ok, I simplify this problem a bit):
  - Three jugs with capacity **a**, **b**, and **c;** and a target capacity **d**
    - The first and the second jug are initially empty, while the third is completely filled with water
      - 1 <= **a, b, c, d** <= 200
    - It is allowed to pour water from one jug into another until either the first one is empty or the second one is full
      - This operation can be performed zero, one or more times.
- Problem: compute the least total amount of water that needs to be poured so that at least one of the jugs contains exactly d liters of water

UVa 10603 – Fill (2)

- Example: a = 2, b = 3, c = 4, d = 1
  - Initially (j1 = 0, j2 = 0, j3 = 4)
  - Pour j3 -> j1 (j1 = 2, j2 = 0, j3 = 2)
    - Cost so far 2
  - Pour j1 -> j2 (j1 = 0, j2 = 2, j3 = 2)
    - Cost so far 2 + 2 = 4
  - Pour j3 -> j1 (j1 = 0, j2 = 3, **j3 = 1**)
    - Cost so far 2 + 2 + 1 = 5 (done)
  - But this is not the best… See next slide

UVa 10603 – Fill (3)

- Example: a = 2, b = 3, c = 4, d = 1
  - Initially (j1 = 0, j2 = 0, j3 = 4)
  - Pour j3 -> j2 (j1 = 0, j2 = 3, **j3 = 1**)
    - Cost so far 3 (done)
  - This is the best that we can do
- No graph in the problem description?

UVa 10603 – Fill (4)

- Slide 4 of this problem is hidden ☺
  - To be discussed in during the recitation, so don’t skip it ☺

Quiz 2 Material: Up to Here ☺

- Any questions for this Quiz 2 next week?
  - You can ask me now or later on Tuesday
  - Note that questions in Quiz 2 arrangement are split between myself (about 50%) and Dr Seth (the rest)