Friday, 23 March 2018

LAB DEMO 07

PS3 Debrief – Common Mistakes (1)

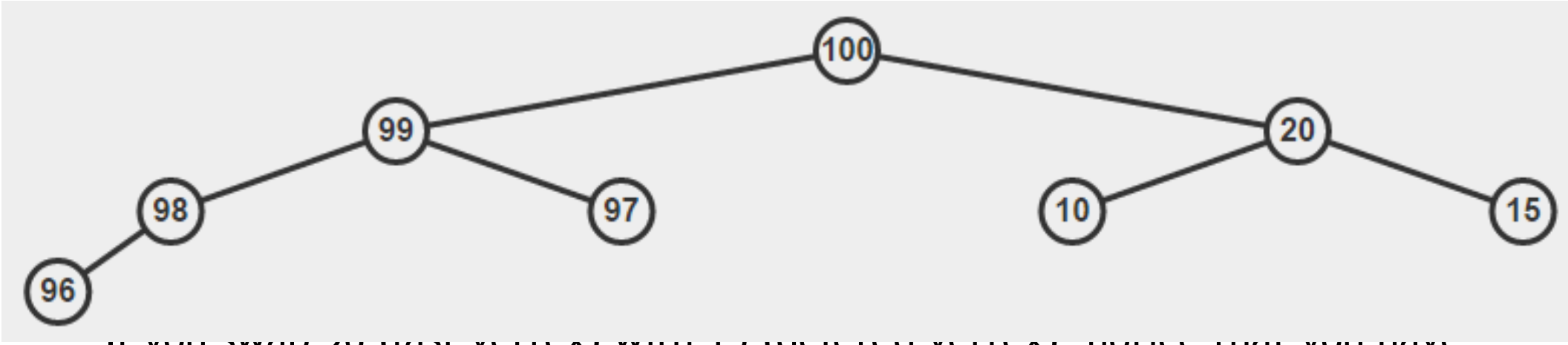
Typical common mistakes:

- TLE in C: Using **O(n)** like method to search the entire PQ of **n** elements for a woman of a certain name

PS3 Debrief – Common Mistakes (2)

Typical common mistakes:

- WA in C: Forgot this case, see below, GiveBirth(15)



If you swap 96 (last vertex) with 15 (deleted vertex), notice that you have to do ShiftUp now, not just ShiftDown like ExtractMax :O

But the easiest implementation is IncreaseKey(15) to INF (e.g. 101 for this PS),
then ExtractMax() ☺️

PS3 Debrief – Our Answers (1)

The expected solution for PS3 Subtask III

- Easiest: Use more than one bBSTs (eh? bBST??)
  - One bBST to map woman name to dilation and arrival time
  - Another bBST to emulate the PQ
  - You can simply use map/set (Week08 topic) to solve PS3… :O :O :O
    - ArriveInHospital() is a simple bBST insertion
    - Query() is a simple bBST FindMax() operation
    - GiveBirth() is simple: search the woman and remove her
    - IncreaseKey(): search the woman, delete her old data, reinsert her new data :O…

PS3 Debrief – Our Answers (2)

- Still easy: Lazy update using PQ (advanced topic, see CP3 page 148-149 or CP3.17b page 161-162), likely very few of you use this…
- Longest to code, which most of you do due to the lesson plan: Write your own Binary Heap class to do UpdateKey and Extract(any_pos) **(remember that this may entail calling either shiftUp and/or shiftDown to fix the binary heap property—a common mistake),** then use unordered_map (Week07 topic) or map (Week08 topic) to map woman name to index

PS4 – The Baby Names Problem

Subtask I/Very Easy:

- How many names start with a certain *letter*?
- Be careful of corner case with START and END
- Can be easily solved with Subtask II code too

Subtask II/Very Easy, with C++ STL… as explained last week

- How many names start with a certain *prefix*?

Subtask III+IV/Tedious, and 0 point somehow :O:

- Subtask II+IV have the same test data, but stricter TL of 1s
- Subtask III has no RemoveSuggestion

Easy Solution for PS4 Subtask I+II

There is one method in Java TreeSet (and TreeMap) that can be **very useful** for PS4 Subtask A+B

- Method **subSet(fromKey, toKey)** in TreeSet
- Method **subMap(fromKey, toKey)** in TreeMap, or
- Near "One liner" solution with just this…

Discussion of which one to use for PS4 Subtask A+B!

- Should be subSet in TreeSet, Q: Why?

"Free" 30+40 = 70 points for those who use this hint

- Plagiarism check is off for A+B, potentially many similar code

What about PS4 Subtask III+IV? (1)

There is one constraint that makes these two subtasks **super tedious**

- You have to **emulate those STL functions**
  - And do faster than O(n) distance calculations…

First of all, you BST has to be **balanced** ☺️

- You have gone through the entire Lecture on Week08 for this
  - Implementing that data structure correctly by yourself is a challenge

What about PS4 Subtask III+IV? (2)

Second, your method has to run in **O(log n)**

- Any non **O(log n)** solution should theoretically will get TLE
- Hint: Scrutinize the “Rank” method that is briefly touched in tutorial tut07

You have 6 more days before PS4 is due to do all these, if you choose to do so ☺️

- Steven guarantees that in CS2040C PE, you CAN just use C++ STL to do the same thing without these “write your own data structure” restriction

Graph DS Review

- VA: https://visualgo.net/en/graphds
- Code: ch2_07_graph_ds.cpp from CP3 book: http://cpbook.net/#downloads

Demo: Graph DS Conversion

By now, you have seen at least 3 graph data structures:

1. Adjacency Matrix (AM)
2. Adjacency List (AL)
3. Edge List (EL)

Each DS is **strong in certain areas** but *weak in another*

- Sometimes we need to convert one DS to another

With 3 different graph DS, there are 6 conversion possibilities: **AM to AL**, AM to EL, AL to AM, AL to EL, EL to AM, and EL to AL

Demo: AM to EL

Today, I will give you a **live demo** on conversion between two graph DS: AM to EL

At home, think of the other possibilities that we have not discussed yet

- Seriously, do not start thinking about this only during Final Assessment in case Steven asks this…

VisuAlgo Training Mode

Today, let's do two

Make sure that you understand the explanation in:

https://visualgo.net/en/bst?slide=1 (until the last slide) and https://visualgo.net/en/graphds?slide=1 (until the last slide too)

Now let's use VisuAlgo Online Quiz training mode to check your basic understanding about BST/AVL and Graph DS on "infinite" number of random questions:

https://visualgo.net/training?diff=Hard&n=5&tl=5&module=bst,avl,graphds

(direct URL problematic now, will debug later; just use main page)

Mock PE 3

Solve CS2040C PE S1 Q1

Before this Lab session runs out (xx.45)!!

Start from this template code (share your repl link)

Gradual hints will be added in few minutes interval