Data Structures
BITL1

Worksheet 13

Fall 2020

10 December 2020

---

1. **Computational complexity:** The following question is about the algorithm below, which takes as input three natural numbers a, b, n.

```
f(a, b, n) :
1  d ← 1
2  let b_k b_{k−1} ··· b_0 be the binary representation of b
3  for i = k, ..., 0:
4    d ← d^2 (mod n)
5    if b_i = 1:
6      d ← da (mod n)
7  return d
```

   (a) Give the values of d as it changes for:

   i. $f(2, 3, 3)$ ii. $f(2, 4, 3)$ iii. $f(2, 5, 3)$

   (b) Suppose that every line takes constant time.

   i. Which inputs does the running time of f depend on?
   ii. What is the running time of f, in Big-O and Big-Ω notation?

   (c) Suppose that lines 4 and 6 take $O(n^2)$ time. In Big-O notation:

   i. What is the running time of f, if $b = 2^k$ for some $k \in \mathbf{N}$?
   ii. What is the running time of f, if $b = 2^k - 1$ for some $k \in \mathbf{N}$?

2. **C++ and the STL:**

   (a) What is the difference between a `class` and a `struct`?

   (b) Suppose that a class `walk` has been defined, containing a method `talk`.

   i. Write the code, outside of the braces `{...};` defining a class `run` derived from `walk` via (public) inheritance.
   ii. Write the code, as the `int main() { ... return 0;}` function, that defines a class `run` called `Gun` and calls its `talk` method.

   (c) What is the difference between `unordered_set` and `unordered_map` in the STL?

3. **Hashing and sorting:**

   (a) Draw what happens when the keys $5, 28, 19, 15, 20, 33, 12, 17, 10$, are insterted into a hash table with hash function $h(k) = k \pmod 9$, with collisions resolved by chaining.

   (b) What is the running time of *quicksort* on an input array of size n, where all the array values are the same?

   (c) What is the running time of *heapsort* on an input array of size n when:

   i. the array is already sorted? ii. the array is sorted in reverse?

4. **Trees:** Recall the *nodes* of a *tree* are either *internal* or *external* (or *leaves*), and they are indexed by *keys*. Each node has a single *parent* and may have one or more *children*. Suppose a tree has n nodes. Answer (a) and (b) in terms of Big-O and Big-Ω.

   (a) What is the running time of inserting a new node into a *binary search tree* (BST)?

   (b) What is the running time of inserting a new node into a *height-balanced* BST (AVL)?

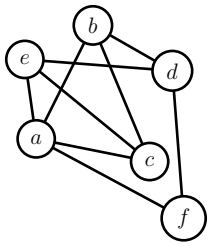
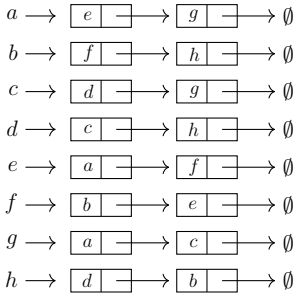
   (c) Draw the *red-black tree* resulting from inserting the keys $5, 16, 22, 45, 2, 10, 18, 30, 50, 12, 1$, in this order, into an empty tree.

   (d) Draw an example of:

      i. a red-black tree that is not an AVL tree

      ii. a red-black tree which after splaying a leaf is not a red-black tree anymore

5. **Undirected graphs:** Complete the table below, where each line is a graph and each column is a way to represent the graph.

graph drawing	adjacency matrix	adjacency list
		
	$\begin{bmatrix} 0&0&0&1&1&1 \\ 0&0&0&1&1&1 \\ 0&0&0&1&1&1 \\ 1&1&1&0&0&0 \\ 1&1&1&0&0&0 \\ 1&1&1&0&0&0 \end{bmatrix}$	
		

6. **Graph algorithms:** Answer (a) and (b) in the context of running time, where n is the number of vertices in the input graph.

   (a) Dijktra's and the Bellman–Ford algorithms find the *shortest path* from a vertex in a weighted directed graph. How are the two different?

   (b) Kruskal's and Prim's algorithms find the *minimum spanning tree* in a weighted undirected graph. How are the two different?

   (c) What type of a graph does the *topological sorting* algorithm take as input?

   (d) What does the *travelling salesperson problem* try to find in an undirected, connected weighted graph?