Discrete Structures

Worksheet 12.2

Spring 2021
March 26

---

1. **Warm up:**

   (a) 3 hats are randomly given to 3 people. What is the probability that nobody gets his/her own hat?

   (b) There are two independent events:
   E_1 has 50% probability.
   E_2 has 20% probability. Find the probability that at least one of the events E_1, E_2 has occured.

   (c) Pick a random number $n \in \{1, \ldots, 40\}$; what is the probability that the fraction $n/40$ is irreducible?

2. Answer the following questions about the Figure 1. Assume that all the disks initially were on Peg 1 and the player wanted to move the whole stack to another peg.

   (a) Which end-state can be achieved faster from the position shown in Figure –fig:w12-hanoi-position: all disks on Peg 2 or all disks on Peg 3?

   (b) How many steps have already been completed to reach the current state (from the original state where all disks are on the Peg 1). How many steps are still necessary to reach the end-state?

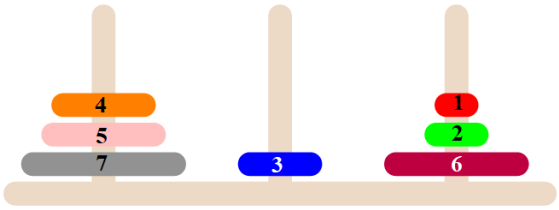

Figure 1: Some Hanoi Tower Position

3. Use the Master theorem to give tight asymptotic bounds for the following recurrences.

   (a) $T(n) = 2T(n/4) + 1$
   (b) $T(n) = 2T(n/4) + \sqrt{(}n)$
   (c) $T(n) = 2T(n/4) + n$
   (d) $T(n) = 2T(n/4) + n^2$

4. Here is a "weighted" variant of a Tower of Hanoi. You still have exactly 7 disks as in Figure 1. Moving Disk #1 costs 1 EUR, moving Disk #2 costs 2 EUR, moving Disk #3 costs 4 EUR, and so on. Moving Disk #7 costs 64 EUR.

   (a) Assume that your task is to move all disks from Peg 1 to some other peg. For every disk size count the times it has to be moved.

   (b) Calculate the total cost to move all disks from one peg to another.

   (c) Assume that we have a robot that supports the following procedure:

   ```
   move(n, pegFrom, pegTo)
   ```

This procedure assumes that all the disks numbered $1, 2, \ldots, n$ are already on Peg number `pegFrom`. The device moves them all to the Peg `pegTo`. The procedure is recursive (and it may call another procedure `moveDisk` to move one specific disk, if it is at the top). Draw one more layer to the Figure 2 and for each node write the associated costs.

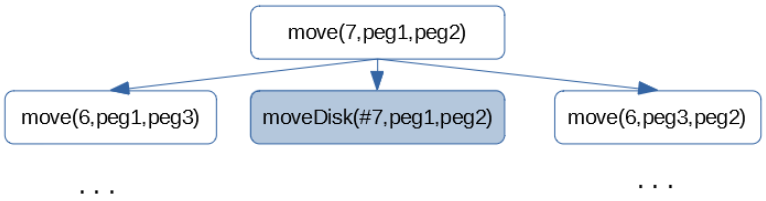

Figure 2: The tree of recursive calls

5. **Divide-and-Conquer algorithms.** Imagine that somebody has invented a new operation $a \otimes b$ for some objects a, b (both a, b have the same size n). Assume that s/he knows how to express $a \otimes b$ using 7 operations $a_i \otimes b_i$ (where $i = 1, 2, \ldots, 7$, and all a_i, b_i have size $n/2$, i.e. half the size of the original operands a, b). (*We do not care, what the operation $\otimes$ does; but we know that we can compute it for arguments a, b of length 1 in constant time; it is therefore easy for short arguments.*)

   Find the best Big-O-Notation estimate for the time needed to compute $a \otimes b$, if a, b are both of size n.

   (A) $O(n^2)$
   (B) $O(n^2 \log n)$
   (C) $O(n^{2.646})$
   (D) $O(n^{2.808})$
   (E) $O(n^3)$
   (F) $O(n^3 \log n)$

6. Assume that there is a real-estate dealer; she wants to get the very best deal between $N = 100$ proposals. Every proposal p_i can be evaluated by a number $v(p_i)$ (a larger number means a better deal). Assume that all the values $v(p_i)$ are different; and the proposals $p_1, \ldots, p_N$ arrive in a random order (every permutation is equally likely). Her strategy is as follows:

   - Evaluate the first k proposals (for some fixed number $k \in [0; N-1]$), but reject them.
   - Evaluate the subsequent proposals $p_{k+1}, \ldots, p_N$ and select the first proposal p_j which satisfies $v(p_j) > \max(v(p_1), v(p_2), \ldots, p(p_k))$, if it exists.

   Answer the following questions:

   (a) What are the possible outcomes of this algorithm; for what permutations does this strategy lead to the best deal?
   (b) What is the probability to find the best proposal for $k = 0$, $k = 1$, $k = N - 1$. (You can replace $N = 100$.)
   (c) What is the optimal value of the parameter k that gives the best chance to find the best proposal with the algorithm given above? (You may need a computer simulation to find out.)