CPSC 320 2016W2: Quiz 4 Pre-Release Information

February 24, 2017

This week's quizzes will follow up on the problem domains described below. It's worth spending a few minutes reading and understanding each domain before your tutorial!

Note that if you collaborate on understanding this information, you should follow the academic conduct guidelines. Among other rules, take down the GradeScope student #s or account names of collaborators for acknowledgement, but take **no other notes** away from those collaborators!

NOTE: The names of problems (on the quiz) subproblems are often intended to be fun and are completely irrelevant. Feel free to ignore them here and on the quiz!

Please, remember your GradeScope # when coming to the tutorial quiz.

Reference (will be available on all quizzes that need it)

The Master Theorem

For a recurrence like $T(n) = aT(\frac{n}{b}) + f(n)$, where $a \geq 1$ and $b > 1$, the Master Theorem states three cases:

1. If $f(n) \in O(n^c)$ where $c < \log_b a$ then $T(n) \in \Theta(n^{\log_b a})$.
2. If for some constant $k \geq 0$, $f(n) \in \Theta(n^c (\log n)^k)$ where $c = \log_b a$, then $T(n) \in \Theta(n^c (\log n)^{k+1})$.
3. If $f(n) \in \Omega(n^c)$ where $c > \log_b a$ **and** $af(\frac{n}{b}) \leq kf(n)$ for some constant $k < 1$ and sufficiently large n, then $T(n) \in \Theta(f(n))$.

1 I'm a Lumberjack (And I'm Okay)

Your task is to design algorithms to solve the following problems. For full credit, your algorithm must run in logarithmic time.

2 Tiles and Tribulations

You're given a grid of size $n \times n$, where $n = 2^k$ for some $k \geq 1$, with one cell missing.

An L-tile is three squares that form an L shape (i.e., a 2×2 square with one square missing).

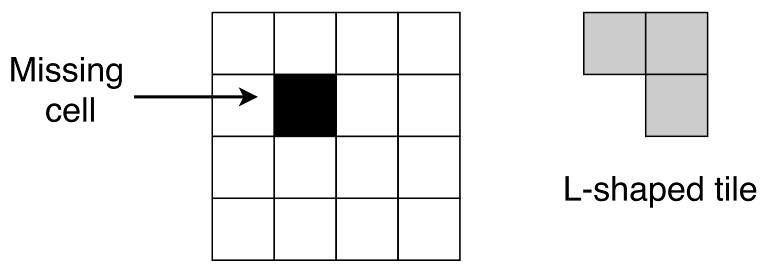

Figure 1: A 4×4 board, with a missing cell at position $(2, 3)$ (using 1-based indexing from bottom left).

.

3 Greed is Good, But Conquest is Better

Our UBC student from quiz 3 has finished her co-op position at Greedy Bank. Disillusioned by Greedy Bank's predatory debit-reordering schemes, she's moved on to bigger and better things at – you guessed it – a different bank.

At DC Bank, overdraft fees are charged over an interval of transactions by multiplying the number of transactions by the *minimum amount by which the account is overdrawn* over that same period. For example, if Customer A's account balances were $[-20, -50, -70, -10]$, Customer A would be charged $4 * \$10 = \$40$. If Customer B's account balances were $[-20, -40, +10, -10]$, he would not be charged anything, since at the third transaction his overdraft amount was zero (as he had a positive balance).

We can define this mathematically as follows: for an size n array A of account balances, the overdraft fee is $f(A) = \max\{0, -n * \max(A[1], A[2], \ldots, A[n])\}$. We can also charge overdraft fees for a particular *interval* of A: starting at index i for an interval of d transactions, the overdraft fee is $f(i, d) = \max\{0, -d * \max(A[i], A[i+1], \ldots, A[i+d-1])\}$.

4 Debug-and-Conquer

No pre-reading for this question; but, make sure you know what a minimum spanning tree is.