Martin Ziegler
Sewon Park, Junhee Cho, Ivan Koswara

Issued on Apr.4, 2018
Solutions due: Apr.13, 2018

CS500

Spring 2018, Assignment #3

PROBLEM 6 (1+3+3+3P) **:**
Consider the following algorithm, regardless of its purpose:

> *Given an n-tuple $(b_1,\ldots,b_n)$ of bits, scan for the (index j of the) first bit that is non-zero. If all are zero, count to 2^n-1. Then stop.*

a) Analyze the worst-case cost of this algorithm, asymptotically as $n \to \infty$.

b) Analyze the average-case cost of this algorithm, asymptotically as $n \to \infty$.

c) Consider a binary counter, initialized with zero and supporting *two* operations: increment and decrement; decrement(0)=0.
Analyze the amortized number of bit-flips. Prove that your asymptotic estimate is tight.

d) Consider a binary counter initialized with any ℓ-bit number and then incremented n times. Devise a potential function to prove that the amortized number of bit-flips is $\mathcal{O}(1+\ell/n)$.

PROBLEM 7 (5+5P)**:**

a) Draw the Fibonacci Heap (with marks) resulting from the following sequence of operations:

`create`(9), `insKey`(8), `insKey`(7), `extrMin`(), `insKey`(6),
`insKey`(5), `insKey`(4), `extrMin`(), `delKey`(6), `insKey`(3),
`insKey`(2), `insKey`(1), `extrMin`(), `delKey`(3), `insKey`(3).

b) Draw the final disjoint forest (including ranks) resulting from the following sequence of operations according to *Lazy Union-by-Rank*[*] and *Path Compression*:

`make`(1), `make`(2), `make`(3), `make`(4), `make`(5), ..., `make`(8), `make`(9),
`union`(1,2), $x :=$ `find`(1), `union`(x,3), `union`(x,4), `union`(x,5), `union`(6,7),
`union`(8,9), `union`(`find`(6),`find`(8)), `union`(`find`(7),x), `find`(3)

[*]When taking the union of two trees of same tank, let ties be broken s.t. the *second* one becomes *child* of the first.