# Foundations of Algorithms
# Homework #5

All members of the collaboration group are expected to participate fully in solving collaborative problems, and peers will assess performance at the end of the assignment. Note, however, that each student is required to write up their solutions individually. Common solution descriptions from a collaboration group will not be accepted. Furthermore, to receive credit for a collaboration problem, each student in the collaboration group must actively and substantially contribute to the collaboration. This implies that no single student should post a complete solution to any problem at the beginning of the collaboration process.

## Problems for Grading

1. [20 points] Write a $\Theta(m + n)$ algorithm that prints the in-degree and the out-degree of every vertex in an $m$-edge, $n$-vertex directed graph where the directed graph is represented using adjacency lists. (Hint: See Figure 2.2 on page 590 of CLRS for a diagram and description of adjacency list representations of graphs.)

2. [10 points] CLRS 9.3-9: Professor Olay is consulting for an oil company, which is planning a large pipeline running east to west through an oil field of $n$ wells. The company wants to connect a spur pipeline from each well directly to the main pipeline along a shortest route (either north or south), as shown in Figure 9.2 in the textbook. Given the $x$- and $y$-coordinates of the wells, how should the professor pick the optimal location of the main pipeline, which would be the one that minimizes the total length of the spurs? Show how to determine the optimal location in linear time, and prove the correctness and linear bound of the algorithm.

3. [40 points] ***Collaborative Problem***–CLRS 5-1: With a $b$-bit counter, we can ordinarily only count up to $2^b - 1$. With R. Morris's *probabilistic counting*, we can count up to a much larger value at the expense of some loss of precision.

   We let a counter value $i$ represent a count of $n_i$ for $i = 0, 1, ..., 2^b - 1$, where the $n_i$ form an increasing sequence of nonnegative values. We assume that the initial value of the counter is 0, representing a count of $n_0 = 0$. The **Increment** operation works on a counter containing the value $i$ in a probabilistic manner. If $i = 2^b - 1$, then the operator reports an overflow error. Otherwise, the **Increment** operator increases the counter by 1 with probability $1/(n_{i+1} - n_i)$, and it remains unchanged with probability $1 - 1/(n_{i+1} - n_i)$.

   If we select $n_i = i$ for all $i \geq 0$, then the counter is an ordinary one. More interesting situations arise if we select, say, $n_i = 2^{i-1}$ for $i > 0$ or $n_i = F_i$ (the $i$th Fibonacci number–See Section 3.2 in the text).

   For this problem, assume that $n_{2^b-1}$ is large enough that the probability of an overflow error is negligible.

   (a) [20 points] Prove that the expected value represented by the counter after $n$ **Increment** operations have been performed is exactly $n$.

   (b) [20 points] The analysis of the variance of the count represented by the counter depends on the sequence of the $n_i$. Let us consider a simple case: $n_i = 100i$ for all $i \geq 0$. Determine the variance in the value represented by the register after $n$ **Increment** operations have been performed.

4. [30 points] ***Collaborative Problem***– In this problem we consider two stacks, A and B, manipulated using the following operations ($n$ denotes the size of A and $m$ the size of B):

   - $PushA(x)$: Push element $x$ on stack A.
   - $PushB(x)$: Push element $x$ on stack B.
   - $MultiPopA(k)$: Pop $min(k, n)$ elements from A.
   - $MultiPopB(k)$: Pop $min(k, m)$ elements from B.
   - $Transfer(k)$: Repeatedly pop an element from A and push it on B, until either $k$ elements have been moved or A is empty.

Assume that A and B are implemented using doubly-linked lists such that $PushA$ and $PushB$, as well as a single pop from A or B, can be performed in $O(1)$ time worst-case.

(a) [5 points] What is the worst-case running time of the operations $MultiPopA$, $MultiPopB$, and $Transfer$?

(b) [25 points] Define a potential function $\Phi(n, m)$ and use it to prove that the operations have amortized running time $O(1)$.