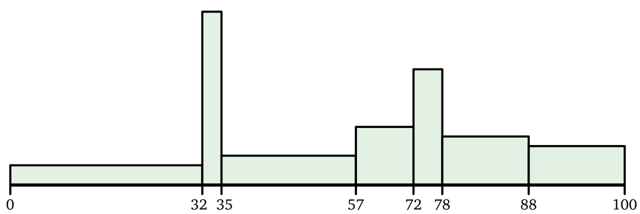
# CS 473 ✧ Spring 2020

# Homework 3

Due Wednesday, February 19, 2020 at 9pm

---

1. ***[Extra credit]*** Suppose we want to visualize a large set $S$ of values—for example, grade-point averages for every student who ever attended UIUC—using a variable-width histogram. To construct a histogram, we choose a sorted sequence of ***breakpoints*** $b_0 < b_1 < \cdots < b_k$, such that every element of $S$ lies between $b_0$ and $b_k$. Each interval $[b_{i-1}, b_i)$ between two consecutive buckets is called a *bin*. Any histogram includes a rectangle for each bin, whose height is the number of elements of $S$ that lie inside that bin.



A variable-width histogram with seven bins.

Unlike a standard histogram, which requires the intervals to have equal width, we are free to choose the breakpoints arbitrarily. For visualization purposes, it is useful for the *areas* of the rectangles to be as close to equal as possible, so we want the sum of the squares of the areas to be as small as possible.[1] To simplify computation, we require that every breakpoint is an element of the dataset $S$.

More precisely, suppose we are given a sorted array $S[1..n]$ of distinct real numbers and an integer $k$. For any indices $i < j$, let

$$area(i, j) = (j - i) \cdot (S[j] - S[i])$$

denote the area of a single histogram rectangle representing the $j - i$ items in the interval $S[i..j-1]$. A histogram for $S$ is determined by a sorted array $B[0..k]$ of distinct *breakpoint* indices, such that $B[0] = 1$ and $B[k] = n$. We need to choose these breakpoints to minimize the sum of the squared areas of its rectangles:

$$Cost(B) = \sum_{i=1}^{k} \Big(area(B[i-1], B[i])\Big)^2.$$

(a) Define an upper-triangular array $A[1..n, 1..n]$ by setting $A[i, j] = (area(i, j))^2$ if $i < j$ and leaving $A[i, j]$ undefined otherwise. Prove that this array has the Monge property.

A partial array has the Monge property if, for all indices $i < i'$ and $j < j'$ *such that* $A[i, j]$ *and* $A[i', j]$ *and* $A[i, j']$ *and* $A[i', j']$ *are defined*, we have $A[i, j] + A[i', j'] \le A[i, j'] + A[i', j']$. If $A$ is upper-triangular, it suffices to check the Monge condition where $i' = i + 1$ and $j' = j + 1$.

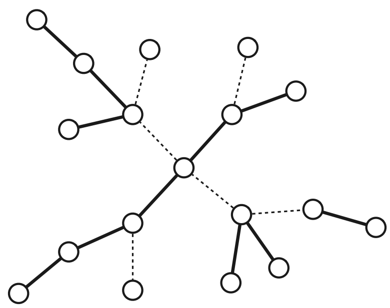
---

[1] As far as I know, this objective has no statistical justification, but it makes the pictures look nice.

(b) Describe an algorithm to find the minimum element in every row of an $n \times n$ *upper triangular* Monge array in $O(n \log n)$ time. (The original SMAWK algorithm requires a full rectangular array.) *[Hint: Use SMAWK as a subroutine.]*

(c) Describe and analyze an algorithm to compute a variable-width histogram with minimum cost for a given set $S$ of data values and a given number $k$ of bins. For full credit, your algorithm should take advantage of parts (a) and (b). *[Hint: You can assume parts (a) and (b) even without a proof.]*

2. Let $T$ be an arbitrary tree—a connected undirected graph with no cycles. Describe and analyze an algorithm to cover the vertices of $T$ with as few disjoint paths as possible. Each vertex of $T$ must lie on exactly one of the paths. (The figure below shows a tree covered by seven disjoint paths, three of which have length zero.)



3. Consider the following non-standard algorithm for shuffling a deck of $n$ cards, initially numbered in order from 1 on the top to $n$ on the bottom. At each step, we remove the top card from the deck and ***insert*** it randomly back into in the deck, choosing one of the $n$ possible positions uniformly at random. The algorithm ends immediately after we pick up card $n-1$ and insert it randomly into the deck.

(a) Prove that this algorithm uniformly shuffles the deck, meaning each permutation of the deck has equal probability. *[Hint: Prove that at all times, the cards below card $n-1$ are uniformly shuffled.]*

(b) What is the *exact* expected number of steps executed by the algorithm? *[Hint: Split the algorithm into phases that end when card $n-1$ changes position.]*

$\pi$. **[Warmup only; do not submit solutions]**

After sending his loyal friends Rosencrantz and Guildenstern off to Norway, Hamlet decides to amuse himself by repeatedly flipping a fair coin until the sequence of flips satisfies some condition. For each of the following conditions, compute the *exact* expected number of flips until that condition is met.

(a) Hamlet flips heads.

(b) Hamlet flips both heads and tails (in different flips, of course).

(c) Hamlet flips heads twice.

(d) Hamlet flips heads twice in a row.

(e) Hamlet flips heads followed immediately by tails.

(f) Hamlet flips more heads than tails.

(g) Hamlet flips the same number of heads and tails.

(h) Hamlet flips the same positive number of heads and tails.

(i) Hamlet flips more than twice as many heads as tails.

*[Hint: Be careful! If you're relying on intuition instead of a proof, you're probably wrong.]*