CS 573: Algorithms, Fall 2014

Homework 5, due Monday, December 1, 23:59:59, 2014

Version 1.0

Neatly print your name(s), NetID(s) on each submitted question. Remember that you have to submit each question on a separate page. each student should submit his own homework. If you are on campus, submit the homework by submitting it in the homework boxes in the basement of SC.

> "Is there anything in the Geneva Convention about the rules of war in peacetime?" Stanko wanted to know, crawling back toward the truck.
> "Absolutely nothing," Caulec assured him. "The rules of war apply only in wartime. In peacetime, anything goes."
> – Gasp, Romain Gary

Required Problems

1. Sorting networks stuff (20 PTS.)

(A) (2 PTS.) Prove that an n-input sorting network must contain at least one comparator between the ith and $(i+1)$st lines for all $i = 1, 2, ..., n-1$.

(B) (10 PTS.) Prove that in a sorting network for n inputs, there must be at least $\Omega(n \log n)$ gates. For full credit, your answer should be short, and self contained (i.e., no reduction please).
[As an exercise, you should think why your proof does not imply that a regular sorting algorithm takes $\Omega(n \log n)$ time in the worst case.]

(C) (3 PTS.)
Suppose that we have $2n$ elements $\langle a_1, a_2, ..., a_{2n} \rangle$ and wish to partition them into the n smallest and the n largest. Prove that we can do this in constant additional depth after separately sorting $\langle a_1, a_2, ..., a_n \rangle$ and $\langle a_{n+1}, a_{n+2}, ..., a_{2n} \rangle$.

(D) (5 PTS.)
Let $S(k)$ be the depth of a sorting network with k inputs, and let $M(k)$ be the depth of a merging network with $2k$ inputs. Suppose that we have a sequence of n numbers to be sorted and we know that every number is within k positions of its correct position in the sorted order, which means that we need to move each number at most $(k-1)$ positions to sort the inputs. For example, in the sequence 3 2 1 4 5 8 7 6 9, every number is within 3 positions of its correct position. But in sequence 3 2 1 4 5 9 8 7 6, the number 9 and 6 are outside 3 positions of its correct position.
Show that we can sort the n numbers in depth $S(k)+2M(k)$. (You need to prove your answer is correct.)

2. Computing Polynomials Quickly (10 PTS.)

In the following, assume that given two polynomials $p(x), q(x)$ of degree at most n, one can compute the polynomial remainder of $p(x)$ mod $q(x)$ in $O(n \log n)$ time. The ***remainder*** of $r(x) = p(x) \bmod q(x)$ is the unique polynomial of degree smaller than this of $q(x)$, such that $p(x) = q(x) * d(x) + r(x)$, where $d(x)$ is a polynomial.
Let $p(x) = \sum_{i=0}^{n-1} a_i x^i$ be a given polynomial.

(A) (2 PTS.) Prove that $p(x) \bmod (x - z) = p(z)$, for all z.
(B) (2 PTS.) We want to evaluate $p(\cdot)$ on the points $x_0, x_1, \ldots, x_{n-1}$. Let

$$P_{ij}(x) = \prod_{k=i}^{j} (x - x_k)$$

and

$$Q_{ij}(x) = p(x) \bmod P_{ij}(x).$$

Observe that the degree of Q_{ij} is at most $j - i$.
Prove that, for all x, $Q_{kk}(x) = p(x_k)$ and $Q_{0,n-1}(x) = p(x)$.
(C) (4 PTS.) Prove that for $i \leq k \leq j$, we have

$$\forall x \quad Q_{ik}(x) = Q_{ij}(x) \bmod P_{ik}(x)$$

and

$$\forall x \quad Q_{kj}(x) = Q_{ij}(x) \bmod P_{kj}(x).$$

(D) (4 PTS.) Given an $O(n \log^2 n)$ time algorithm to evaluate $p(x_0), \ldots, p(x_{n-1})$. Here $x_0, \ldots, x_{n-1}$ are n given real numbers.

3. Linear time Union-Find. (20 PTS.)

(A) (2 PTS.) With path compression and union by rank, during the lifetime of a Union-Find data-structure, how many elements would have rank equal to $\lfloor \lg n - 5 \rfloor$, where there are n elements stored in the data-structure?
(B) (2 PTS.) Same question, for rank $\lfloor (\lg n)/2 \rfloor$.
(C) (4 PTS.) Prove that in a set of n elements, a sequence of n consecutive FIND operations take $O(n)$ time in total.
(D) (4 PTS.) Write a non-recursive version of FIND with path compression.
(E) (4 PTS.) Show that any sequence of m MAKESET, FIND, and UNION operations, where all the UNION operations appear before any of the FIND operations, takes only $O(m)$ time if both path compression and union by rank are used.
(F) (4 PTS.) What happens in the same situation if only the path compression is used?

4. Naive. (10 PTS.)

We wish to compress a sequence of independent, identically distributed random variables $X_1, X_2, \ldots$. Each X_j takes on one of n values. The ith value occurs with probability p_i, where $p_1 \geq p_2 \geq \ldots \geq p_n$. The result is compressed as follows. Set

$$T_i = \sum_{j=1}^{i-1} p_j,$$

and let the ith codeword be the first $\lceil \lg(1/p_i) \rceil$ bits (in the binary representation) of T_i. Start with an empty string, and consider X_j in order. If X_j takes on the ith value, append the ith codeword to the end of the string.

(A) Show that no codeword is the prefix of any other codeword.

(B) Let Z be the average number of bits appended for each random variable X_j. Show that

$$\mathbb{H}(X_j) \leq z \leq \mathbb{H}(X_j) + 1.$$

5. Codification. (20 PTS.)

Arithmetic coding is a standard compression method. In the case when the string to be compressed is a sequence of biased coin flips, it can be described as follows. Suppose that we have a sequence of bits $X = (X_1, X_2, \ldots, X_n)$, where each X_i is independently 0 with probability p and 1 with probability $1 - p$. The sequences can be ordered lexicographically, so for $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, we say that $x < y$ if $x_i = 0$ and $y_i = 1$ in the first coordinate i such that $x_i \neq y_i$. If $z(x)$ is the number of zeroes in the string x, then define $p(x) = p^{z(x)}(1-p)^{n-z(x)}$ and

$$q(x) = \sum_{y<x} p(y).$$

(A) Suppose we are given $X = (X_1, X_2, \ldots, X_n)$. Explain how to compute $q(X)$ in time $O(n)$ (assume that any reasonable operation on real numbers takes constant time).

(B) Argue that the intervals $\Big[q(x), q(x) + p(x)\Big)$ are disjoint subintervals of $[0, 1)$.

(C) Given (A) and (B), the sequence X can be represented by any point in the interval $I(X) = \Big[q(X), q(X) + p(X)\Big)$. Show that we can choose a codeword in $I(X)$ with $\lceil \lg(1/p(X)) \rceil + 1$ binary digits to represent X in such a way that no codeword is the prefix of any other codeword.

(D) Given a codeword chosen as in (C), explain how to decompress it to determine the corresponding sequence $(X_1, X_2, \ldots, X_n)$.

(E) (Extra credit.) Using the Chernoff inequality, argue that $\lg(1/p(X))$ is close to $n\mathbb{H}(p)$ with high probability. Thus, this approach yields an effective compression scheme.

6. Entropy stuff. (20 PTS.)

(A) (5 PTS.) *Maximizing Entropy*
Consider an n-sided die, where the ith face comes up with probability p_i. Show that the entropy of a die roll is maximized when each face comes up with equal probability $1/n$.

(B) (5 PTS.) *Extraction to the limit,*
We have shown that we can extract, on average, at least $\lfloor \lg m \rfloor - 1$ independent, unbiased bits from a number chosen uniformly at random from $\{0, \ldots, m-1\}$. It follows that if we have k numbers chosen independently and uniformly at random from $\{0, \ldots, m-1\}$ then we can extract, on average, at least $k \lfloor \lg m \rfloor - k$ independent, unbiased bits from them. Give a better procedure that extracts, on average, at least $k \lfloor \lg m \rfloor - 1$ independent, unbiased bits from these numbers.

(C) (2 PTS.) Assume you have a (valid) prefix code with n codewords, where the ith codeword is made out of ℓ_i bits. Prove that

$$\sum_{i=1}^{n} \frac{1}{2^{l_i}} \leq 1.$$

(D) (2 PTS.) Let $S = \sum_{i=1}^{10} 1/i^2$. Consider a random variable X such that $\mathbf{Pr}[X = i] = 1/(Si^2)$, for $i = 1, \ldots, 10$. Compute $\mathbb{H}(X)$.
See

(E) (2 PTS.) Let $S = \sum_{i=1}^{10} 1/i^3$. Consider a random variable X such that $\mathbf{Pr}[X = i] = 1/(Si^3)$, for $i = 1, \ldots, 10$. Compute $\mathbb{H}(X)$.

(F) (2 PTS.) Let $S(\alpha) = \sum_{i=1}^{10} 1/i^{\alpha}$, for $\alpha > 1$. Consider a random variable X such that $\mathbf{Pr}[X = i] = 1/(S(\alpha)i^{\alpha})$, for $i = 1, \ldots, 10$. Prove that $\mathbb{H}(X)$ is either increasing or decreasing as a function of α (you can assume that α is an integer).

A quick

1	2.92897	2.87645
2	1.54977	1.78359
3	1.19753	0.929465
4	1.08204	0.477162
5	1.03691	0.250697
6	1.01734	0.134125
7	1.00835	0.0724193
8	1.00408	0.0392272
9	1.00201	0.0212432
10	1.00099	0.0114802

(G) (2 PTS.) The ***conditional entropy*** $\mathbb{H}(Y|X)$ is defined by

$$\mathbb{H}(Y|X) = \sum_{x,y} \mathbf{Pr}[(X = x) \cap (Y = y)] \lg \frac{1}{\mathbf{Pr}[Y = y|X = x]}.$$

If $Z = (X, Y)$, prove that

$$\mathbb{H}(Z) = \mathbb{H}(X) + \mathbb{H}(Y|X).$$