CSE 101 Homework 4

Winter 2022

This homework is due on gradescope Friday February 18th at 11:59pm on gradescope. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in \LaTeX is recommended though not required.

**Question 1** (Hadamard Matrices, 25 points). The $n^{th}$ Hadamard matrix is a $2^n \times 2^n$ matrix of some importance. It can be defined recursively by

$$H_0 = \begin{bmatrix} 1 \end{bmatrix}, \quad H_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}.$$  

Give an algorithm that given a $2^n$-dimensional vector $v$ computes $H_nv$ in time $O(n^2n)$. 

Note: Given that $N = 2^n$ is the dimension of $H_n$ (and the size of the problem), this algorithm runs in nearly linear $O(N \log(N))$ time. Also, this is a reasonable analogue of the Fast Fourier Transform algorithm with some similar applications.

**Question 2** (Bird Keeping, 30 points). Aiden has $n$ birds and $n$ cages to keep them in. Each cage has a positive real numbered size and each bird has a positive real size requirement. Aiden wants to assign as many of his birds as possible each to its own cage so that the size requirement of each bird is no larger than the size of the cage it is assigned to. Give an algorithm that given the lists of cage sizes and bird size requirements determines the maximum number of birds that can be successfully assigned this way.

For full credit, your algorithm should run in time $O(n \log(n))$ or better.

**Question 3** (Least Recently Used, 45 points). Although the Furthest In The Future protocol is guaranteed to produce the optimal answer in the caching problem, it has the issue that implementing it requires that one know the entire sequence of memory lookups ahead of time, which is often not the case. An often more practical protocol is the Least Recently Used (LRU) system, whereby when there is a cache miss, the new item replaces the item currently in cache that was used least recently.

(a) Unfortunately, LRU can often be far from the optimal protocol. Show that for any integer $k$ and real number $\epsilon > 0$, there is a sequence $S$ of memory accesses so that if the LRU protocol is run on the sequence $S$ with a cache of size $k$, the number of cache misses is more than $(k - \epsilon)$ times as large as the optimal schedule for $S$. [15 points]

(b) However, LRU does have some nice properties. For example, suppose that LRU is being run on some sequence of memory accesses with a cache of size $k$. Furthermore, suppose that there is some time period during which our memory accesses only requests $k$ different memory locations. Show that during this time period that LRU makes at most $k$ cache misses. [15 points]

(c) Although LRU may be far from optimal, there is a sense in which it is competitive. Show that for any sequence $S$ of memory accesses and any positive integer $k$ that LRU run on $S$ with a cache of size $2k$ makes at most twice as many cache misses as the best protocol for executing $S$ on a cache of size only $k$. [15 points]

**Question 4** (Extra credit, 1 point). Approximately how much time did you spend working on this homework?