CIS 786 Homework 3 Solution Outline

Problem 1. (60 points)
Separate response will be given.

Problem 2. (20 points)
We use Valiant’s lower bound argument.

Let us have \( n \) keys whose maximum \( \text{MAX}_n \) we must establish in two rounds using \( p \) processors. Let us assume that \( p \geq n \), since otherwise we can use extra processors for free.

The number of parallel comparisons that can be performed in the first round is \( p \). We form a graph with vertices the keys and edges the comparisons established. There is in the formed graph an independent set of size at least \( P = n^2/(2p + n) \) by Turan’s theorem. We can force a second round on the algorithm by fixing these \( P \) keys so that they become candidates for the maximum.

In the second round concentrate on these \( P \) keys only. A number of \( p \) parallel comparisons can determine the Maximum of the \( P \) keys (and of course of the original \( n \) keys) as long as \( p \geq P^2 \geq \binom{P}{2} \) (we use the "CRCW" or the 1-round algorithm described in class). We thus have

\[
p \geq P^2 \geq \binom{P}{2} = \frac{n^2}{2p + n}\\p(2p + n)^2 \geq n^4\\p(2p + p)^2 \geq p(2p + n)^2 \geq n^4\\p(2p + p)^2 \geq n^4\\9p^3 \geq n^4\\p = \Omega\left(n^{4/3}\right).
\]

Problem 3. (20 points)
We start with the description in the notes and then, we continue it.

Consider \( n \) keys. Split them into \( n/3 \) groups of 3 keys each. For each group we can determine the MAX in \( 3(3 - 1)/2 = 3 \) comparisons using 3 procs per group. Total number of processors used is \( n/3 \cdot 3 = n \). Thus we are left with determining the max of \( n/3 \) keys.

Take the \( n/3 \) Maxima, and split them into groups of 7. We have \( n/(3 \cdot 7) \) groups of 7 keys. Each group requires \( 7(7 - 1)/2 = 21 \) processors to find the MAX of the group. Total number of processors used is \( n/21 \cdot 21 = n \), and we can afford to use that many processors. Thus after the second second it suffices to find the max of \( n/21 \) keys to determine the MAX of the original \( n \) keys.

How do we split the \( n/21 \) keys next? What is the pattern?

Say that at some point we end up having \( n/s \) keys. We split them into groups of \( t \) keys so that \( t(t - 1)/2 \) comparisons/processors are assigned per group to determine the group maximum in one step(round).

This would give a total of assigned processors equal to \( (n/(st))t(t - 1)/2 \) which must be at most \( n \) (number of available processors). For this to hold, if we do the math we end up with \( t = 2s + 1 \). Thus we should split the \( n/s \) keys into \( n/(s(2s + 1)) \) groups of \( 2s + 1 \) keys. Or for \( n/21 \) keys, into groups of 43 keys in the following iteration.

What is the pattern? The denominator \( s \) in the next step become \( s(2s + 1) \) or at least \( 2^1 \cdot s^{2^1} \).

In the next step, it becomes \( 2(2s^2)^2 = 2^{1+2}s^{2^2} \). In the following step, it becomes \( 2 \cdot 2^{2+4}s^8 = 2^{1+2+4}s^{2^3} \).
In the $i$-th step it will become $2^{1+2+2^2+\ldots+2^{i-1}}s^{2^i} = 2^{2^i-1}s^{2^i}$.

This denominator becomes greater than $n$, i.e. the number of groups is 1 or less and thus MAX can be determined uniquely for $i$ about $\lg \lg n$. 