PAL: Tutorial, Week 4

Question 1a - CRCW Index of Array Max

1. Performing pairwise check on $i > j$, writing 1 into $M[j]$ where it holds true.
2. Remove duplicates from M.
3. The result is the last remaining index i at which $M[i] = 0$.

	j	1	2	3	4	5
i		7	12	39	15	39
1	7	0	0	0	0	0
2	12	1	0	0	0	0
3	39	1	1	0	1	0
4	15	1	1	0	0	0
5	39	1	1	0	1	0
$M_1 =$		1	1	**0**	1	**0**
$M_2 =$		1	1	**0**	1	1

$Result = 3$

Question 1b - EREW Index of Array Max

- Binary fan-in to find the maximum element
- Binary fan-out broadcast to replicate the maximum element
- Pairwise comparison to find occurrences of the maximum element
- Binary fan-in to find the minimum occurrence ID

Question 2 - CRCW Sort

	$j =$	1	2	3	4	5	6	7	8	9
	$A =$	5	43	12	7	89	99	4	8	9
init.	$W =$	1	1	1	1	1	1	1	1	1
$i = 1$ (5)	$W =$	2	2	2	2	2	2	1	2	2
$i = 2$ (43)	$W =$	2	7	2	2	3	3	1	2	2
$i = 3$ (12)	$W =$	2	7	6	2	4	4	1	2	2
$i = 4$ (7)	$W =$	2	7	6	3	5	5	1	3	3
$i = 5$ (89)	$W =$	2	7	6	3	8	6	1	3	3
$i = 6$ (99)	$W =$	2	7	6	3	8	9	1	3	3
$i = 7$ (4)	$W =$	2	7	6	3	8	9	1	4	4
$i = 8$ (8)	$W =$	2	7	6	3	8	9	1	4	5

The top three rows show the initial data: the index j of each column, the input array A, and the work array W.

Working down the remaining rows, i ranges from 1 to $n-1$ (the value of $A[i]$ is shown in brackets for convenience). On each row, the number at $A[i]$ is compared to all of the values in columns where $j > i$. For example, on the row where $i = 5$, $A[5]$ is compared against $A[6]$, $A[7]$, $A[8]$ and $A[9]$. The numbers that are not compared are shown in **grey**.

Each time $A[i]$ is **strictly greater than** one of the $A[j]$ values, i 'wins' and $W[i]$ is incremented. For example, in the **blue** cell, $A[6]$ (99) was greater than $A[7]$ (4), $A[8]$ (8) and $A[9]$ (9), so $W[6]$ was incremented three times.

Each time $A[i]$ is **equal to or less than** one of the $A[j]$ values, j 'wins' and $W[j]$ is incremented. For example, in the **green** cell, $A[2]$ (43) was less than $A[5]$ (89), so $W[5]$ was incremented.

After $n-1$ passes through the table, the numbers in W represent the correct ordering of the array. The **red** numbers show the final values of W. If B is the output array then $B[W[i]] = A[i]$, or more simply if $W[i] = p$ then $A[i]$ should be in the p^{th} position of the ordered array.

Therefore $B = [4, 5, 7, 8, 9, 12, 43, 89, 99]$.

Question 3 - CRCW Sort With Duplicates

The CRCW sort algorithm from the lecture **does** work with duplicate values:

- For an n-sized input, consider any two duplicate values at positions a and b, where $b = n - m$ and $b = a + k$ and $k > 0$ and $m \geq 0$.
  - i.e. a is before b, there are k items between a and b, and there are m items after b.
- When the item at a is being compared against values in higher positions, it will be compared against k other items before reaching the item at b, giving it k chances to 'win' or 'lose' points.
- Those k items will also be compared against the item at b (but not against the item at a), thus also giving the item at b the same k chances to 'win' or 'lose' points.
- The items in the m positions beyond b will be encountered by comparisons from the items at both a and b, so they do not affect the result. Similarly the items in the $(a-1)$ position before a will be compared to the items at both a and b.
- So far, the items at a and b have both had $k + m + a - 1$ comparisons and will be in equal positions.
- Finally, the item at a is compared against the item at b, but the reverse is not true, thus giving the item at a one extra comparison. The item at a will 'lose' this comparison as it is not strictly bigger than the item at b, therefore placing the item at b one position ahead of the item at a in the output. This is acceptable, as the items at a and b are the same.