CS 337	**Solution to Makeup Quiz 3**	**12/15/03**

Open book and notes.
Max points = 50 Time = 50 min

1. (String Matching)

   (a) (Rabin-Karp algorithm) For $n = 26$, $val(n) = 4$. From the following table, there are 3 possible matches out of which one is successful; so there are 2 failed matches.

text	3	1	4	1	5	8	2	6	5	3	5	9	9	7	9	3
$val(n)$		9	3	8	4	3	5	4	10	9	2	4	0	9	2	5

   (b) (KMP algorithm) Suppose p occurs somewhere in t; consider the first occurrence. Then t is of the form xpy, and pt is of the form $pxpy$. The core of the prefix pxp is p. That is, if p is in t, there is a prefix of pt, of length at least $2 \times p$, whose core is exactly p. Conversely, suppose pt has a prefix z whose core is p. If z is at least $2 \times p$ in length, its suffix which matches p is entirely past p, i.e., within t. The additional condition on length is needed because if $p = aa$ and $t = ab$, then $pt = aaab$, the core of aaa is aa, which is p, though p is not in t.

   (c) Yes. Suppose $v =$ "ab" and $v' =$ "aba". Then $c(v) = \epsilon$, and u has been set to $c(v)$, i.e., ϵ, before this portion of the code is executed. Because $p[\overline{u}]$ and $p[\overline{v}]$ are both "a", $c(v')$ will be set "a".

   (d) (KMP Algorithm) This table shows various values of l. It is possible to terminate the algorithm when $l = 6$, because the text cannot possibly match the pattern at $l = 8$, from length considerations.

l	*index*	0	1	2	3	4	5	6	7	8	9	10	11	12		
	text	b	a	c	b	a	b	a	b	a	a	b	c	b		
0	*pattern*	a	b	a	b	a	c	a								
1	*pattern*		a	b	-	-	-	-	-							
2	*pattern*			a	-	-	-	-	-	-						
3	*pattern*				a	-	-	-	-	-	-					
4	*pattern*					a	b	a	b	a	c	-				
6	*pattern*							a	b	a	b	-	-	-		
8	*pattern*									a	b	-	-	-	-	-

2. (Data Parallel Programming)

(a) (Batcher Merge) First, we show that for any powerlist v, $v \updownarrow v = v \bowtie v$.

$$\begin{array}{ll} & v \updownarrow v \\ = & \{\text{definition of } \updownarrow\} \\ & (v \ \min \ v) \bowtie (v \ \max \ v) \\ = & \{(v \ \min \ v) = v \text{ and } (v \ \max \ v) = v\} \\ & v \bowtie v \end{array}$$

From this result, it is sufficient to prove that $u \ bm \ u = u \updownarrow u$. The proof is by induction on the structure of u.

- $\langle x \rangle \ bm \ \langle x \rangle = \langle x \rangle \updownarrow \langle x \rangle$:

$$\begin{array}{ll} & \langle x \rangle \ bm \ \langle x \rangle \\ = & \{\text{definition of } bm\} \\ & \langle x \rangle \updownarrow \langle x \rangle \end{array}$$

- Given that $p \bowtie q$ is sorted, show that $(p \bowtie q) \ bm \ (p \bowtie q) = (p \bowtie q) \ \updownarrow \ (p \bowtie q)$:

$$\begin{array}{ll} & (p \bowtie q) \ bm \ (p \bowtie q) \\ = & \{\text{definition of } bm\} \\ & (p \ bm \ q) \ \updownarrow \ (q \ bm \ p) \\ = & \{\text{since } p \text{ and } q \text{ are parts of } p \bowtie q \text{ and } p \bowtie q \text{ is sorted}\} \\ & (p \bowtie q) \ \updownarrow \ (p \bowtie q) \end{array}$$

(b) Suppose $p = \langle p_0 \ \cdots \ p_n \rangle$ and $q = \langle q_0 \ \cdots \ q_n \rangle$. Then the i^{th} elements of $ps \ p$, $ps \ q$ and $ps(p+q)$ are, respectively,

$$\begin{array}{l} (ps \ p)_i = (\oplus i : 0 \le j \le i : p_j) \\ (ps \ q)_i = (\oplus i : 0 \le j \le i : q_j) \\ (ps(p \oplus q))_i = (\oplus i : 0 \le j \le i : p_j \oplus q_j) \end{array}$$

Since $\oplus$ is commutative and associative, $(\oplus i : 0 \le j \le i : p_j \oplus q_j) = (\oplus i : 0 \le j \le i : p_j) \oplus (\oplus i : 0 \le j \le i : q_j)$, which is $(ps \ p)_i \oplus (ps \ q)_i$. For the counterexample: let $\oplus$ be string concatenation, $p = \langle 0 \ 1 \rangle$, $q = \langle a \ b \rangle$. Then

$$\begin{array}{l} ps \ p = \langle 0 \ 01 \rangle \\ ps \ q = \langle a \ ab \rangle \\ (ps \ p) \oplus (ps \ q) = \langle 0a \ 01ab \rangle \\ ps(p \oplus q) = ps\langle 0a \ 1b \rangle = \langle 0a \ 0a1b \rangle \end{array}$$