CSE 3500 Algorithms and Complexity

Fall 2014 Exam III – Solutions

1. Let \( A = \{a_1, a_2, \ldots, a_n\} \) and \( B = \{b_1, b_2, \ldots, b_n\} \). Group the processors into \( G_1, G_2, \ldots, G_n \) where each group has \( n \) processors. Group \( G_i \) is assigned the key \( a_i \) (for \( 1 \leq i \leq n \)). \( G_i \) checks if \( a_i \) is repeated in \( B \) by comparing \( a_i \) with every element of \( B \) and taking a Boolean OR. At the end we again use the Boolean OR algorithm to check if any element of \( A \) is repeated in \( B \). The total run time is \( O(1) \).

2. Partition the processors so that there are \( n \) groups \( G_i, 1 \leq i \leq n \), with \( n^2 \) processors in each group. Assign \( k_i \) to \( G_i \). Each \( G_i \) maintains an array \( B \) of size \( n \), one cell for each of the keys in the input sequence. Each cell in \( B \) is initialized to \( \infty \). \( n \) of the processors in \( G_i \) compare \( k_i \) with every input key and set \( B[j] \) to \( k_j \) if \( k_j \) is greater than \( k_i \). Now, find the minimum of all the keys in \( B \). This gives the right neighbor of the key \( k_i \). Since the minimum of \( n \) elements can be found in \( O(1) \) time using \( n^2 \) common CRCW PRAM processors, the run time of the algorithm is \( O(1) \).

3. Let the input sequence be \( X = b_1, b_2, \ldots, b_n \). Form the sequence \( Y = (1, b_1), (2, b_2), \ldots, (n, b_n) \). This sequence can be formed in \( O(\log n) \) time given \( n \) CREW PRAM processors.

   Now define an operator \( \oplus \) as follows: \((i, 1) \oplus (j, b) = (i, 1)\) for any \( i, j \), and \( b \). Also, \((i, 0) \oplus (j, b) = (j, b)\) for any \( i, j \), and \( b \). Clearly, this operator is associative and can be computed in \( O(1) \) time.

   Compute \((1, b_1) \oplus (2, b_2) \oplus \cdots \oplus (n, b_n)\) using a prefix computation. If the result is \((i, o)\) for some \( i \), then it means that there is no \( 1 \) in the sequence \( X \). If the result is \((i, 1)\) for some \( i \), then the position of the leftmost \( 1 \) in \( X \) is \( i \).

   Clearly, the algorithm takes \( O(\log n) \) using \( \frac{n}{\log n} \) CREW PRAM processors.

4. Assign \( q = \frac{m}{\log m} \) CREW PRAM processors for every position \( i \) in the text (for \( 1 \leq i \leq (n - m + 1) \)). Let these \( q \) processors be \( P_1^i, P_2^i, \ldots, P_q^i \).

   for \( i = 1 \) to \( (n - m + 1) \) in parallel do
   for \( j = 1 \) to \( q \) in parallel do
       Processor \( P_j^i \) computes \( b_k^j := \text{"Is } t_{i+k-1} \neq p_k?'\),
       for \( k = (j-1) \log m + 1, (j-1) \log m + 2, \ldots, j \log m \);
   for \( i = 1 \) to \( (n - m + 1) \) in parallel do
       Processors \( P_1^i, P_2^i, \ldots, P_q^i \) collectively compute \( A[i] \) as \( b_1^i + b_2^i + \cdots + b_m^i \)
       using a prefix computation;

   Clearly, the number of processors needed is \( \frac{(n-m+1)m}{\log m} \leq \frac{nm}{\log m} \). The run time of each step is \( O(\log m) \).

5. If \( n \) is the size of the instance of \( \pi_1 \), we can translate this instance into an instance of \( \pi_2 \) in \( n^3 \) time. This implies that the size \( m \) of the corresponding instance of \( \pi_2 \) cannot be more than \( n^3 \). Since the algorithm for \( \pi_2 \) takes \( O(m^{10}) \) time, this run time is no more than \( O((n^3)^{10}) = O(n^{30}) \). Thus the instance of \( \pi_1 \) can be solved in time \( n^3 + O(n^{30}) = O(n^{30}) \).
6. Let $G(V, E)$ be any instance of HCP. We construct an instance of TSP as follows: $G'(V, E')$; $n$ (where $n = |V|$) such that if $(a, b) \in E$, then this edge will have a weight of 1 in $G'$. If $(a, b) \notin E$, then we add the edge $(a, b)$ to $G'$ with a weight of 2. Clearly, this construction takes $O(|V|^2)$ time.

We have to show that $G$ has a Hamiltonian cycle if and only if $G'$ has a tour of weight $\leq n$.

If $G$ has a Hamiltonian cycle, then clearly, this cycle is a tour of weight $n$ in $G'$. Also, if $G'$ has a tour of weight $n$, then it means that each edge in this path has a weight of 1 which in turn means that each of these edges will also be an edge in $G$. As a result, this path in $G'$ corresponds to a Hamiltonian cycle in $G$.

Put together, we infer that HCP polynomially reduces to TSP.