CSE 5500 Algorithms

Fall 2018; Homework 3 Solutions

1. It was shown in class that the maximum of n elements can be found in $O(1)$ time using n^2 common CRCW PRAM processors.

   Consider the case when $\epsilon = \frac{1}{2}$. Divide the elements into groups fo size $\sqrt{n}$. Assign the first $\sqrt{n}$ elements to the first n processors and the second $\sqrt{n}$ elements to the next n processors and so on. The maximum element in each group can be found in $O(1)$ time. At this stage, we have $\sqrt{n}$ elements and $n\sqrt{n}$ processors. Hence, the maximum of these elements can be found in $O(1)$ time. Total time = $O(1)$.

   Next, consider the case when $\epsilon = \frac{1}{3}$. Here, divide the elements into groups of size $n^{1/3}$. Assign the first $n^{1/3}$ elements to the first $n^{2/3}$ processors and the second $n^{1/3}$ elements to the next $n^{2/3}$ processors and so on. The maximum element of each group can be found in $O(1)$ time and using $n^{4/3}$ prceossors the maximum of these maximum elements can be found in $O(1)$ time.

   For the general case, partition the input into groups with n^ϵ elements in each group. Find the maximum of each group assigning $n^{2\epsilon}$ processors to each group. This takes $O(1)$ time. Now the problem reduces to finding the maximum of $n^{1-\epsilon}$ elements. Again, partition the elements with n^ϵ elements in each group and find the maximum of each group. There will be only $n^{1-2\epsilon}$ elements left. Proceed in a similar fashion until the number of remaining elements is $\leq \sqrt{n}$. The maximum of these can be found in $O(1)$ time. Clearly, the run time of this algorithm is $O(1/\epsilon)$. This will be a constant if ϵ is a constant.

2. Assign n^2 processors to each of the input keys. Let G_i be the collection of processors associated with k_i. G_i identifies all the keys in the input that are greater than k_i and then finds the minimum of all these keys in $O(1)$ time. This minimum is the right neighbor of k_i.

   For the randomized algorithm use the fact that we can find the minimum of n keys in $\widetilde{O}(1)$ time using n common CRCW PRAM processors.

3. We can use an array $a[1:n]$ to solve this problem. At the beginning processor 1 sets $ThereAreRepeatedElements := 0$. Assume that we have n arbitrary CRCW PRAM processors. Let the input sequence be $k_1, k_2, \ldots, k_n$. We assign one key per processor. In one parallel write step, for $1 \leq i \leq n$, processor i tries to write i in $a[k_i]$. In the next step processor i reads from $a[k_i]$. If it does not find i there it tries to write a 1 in $ThereAreRepeatedElements$. At the end of this step, we have the result in $ThereAreRepeatedElements$.

4. We know that π_1 polynomially reduces to π_2. Let x be an instance of π_1 with $|x| = n$. We can convert this into an instance x' of π_2 in $O(n^c)$ time (for some constant c). Note that c could be any constant (10, for instance) and we can only say that $|x'| = O(n^c)$ and in fact $|x'|$

could be $\Omega(n^c)$. If $|x'|$ is $\Omega(n^c)$, the run time needed for solving x' will be $O(2^{\sqrt{\Omega(n^c)}})$ which can be asymptotically greater than $2^{\sqrt{n}}$. Thus the given statement is not correct.

5. Use the following algorithm, $\mathsf{Size}(\mathit{Graph}\ G)$ -

   for $i := |V|$ to 0 **do**
       if $\mathsf{CLQ}(i) =$ **yes then**
           output i
           quit
   end

   Note that we increase the runtime of the CLQ algorithm, by a factor of $|V|$, yet maintaining it polynomial.