Advanced Data Structures (spring 2007)

Exercise 5 (Wed 25.4, 12-14, C221)

   Design a cache-oblivious algorithm for computing the Euler-tour of a tree (see Wiki page on lowest common ancestor query).
   
   - The input is an array of the tree edges, where each edge is the pair \(\langle\text{parentid}, \text{childid}\rangle\).
   - The output is an array of node identifiers in the Euler-tour order (see array \(E\) on the LCA Wiki page).
   
   The algorithm should use \(O\left(\frac{N}{B}\log_{M/B}\frac{N}{B}\right)\) memory transfers. (Hint: Construct the Euler-tour first as a (static) linked list and apply list ranking.)

2. Quicksort in external memory.
   a) Write a recursive formula describing the number of memory transfers made by the cache-oblivious median-quicksort algorithm described at lecture.\(^1\) Solve it to show that it gives the same result as the analysis in the Wiki page.
   b) Change the quicksort algorithm so that it works optimally in the cache-aware model. You can assume an oracle that gives you \(k\)-quantiles, i.e., \(k-1\) partitioning elements to partition the data items into \(k\) equal size disjoint buckets so that all the items in one bucket precede all the items in the next bucket. Value \(k\) must be smaller than the cache size \(M\).

   The cache-oblivious model assumes optimal replacement strategy and full associativity. Find out what replacement strategy and associativity mean and what they are in a modern processor of your choice.


---

\(^1\)Notice that the description has been modified after the lecture.