Searching in a Sorted Sequence in Parallel

Dr.N.Sairam & Dr.R.Seethalakshmi
School of Computing,
SASTRA Univeristy,
Thanjavur-613401.

Contents

1. Searching in a Sorted Sequence in Parallel

Computer systems are often used to store large amounts of data from which individual records must be retrieved according to some search criterion. Thus the efficient storage of data to facilitate fast searching is an important issue. Everyday all search for something or other. Users generally browse the net for contents. This is a typical search handled by the search engine. In the case of organized data, we search for the data set. Searching

may occur both in a sorted sequence or unsorted or random sequence. In this section the searching on a sorted sequence in parallel is discussed.

The algorithm consists of a array where there are n elements. Let S be the array in which the data is stored. Let p be the number of processors used for searching. Generally p<n.

Algorithm Parallelsearch(S,n,y)

Input:- A sorted array S=x_1,x_2,x_3 …xn consisting of n elements.

A query element y.

Output:- y and its position in the corresponding sub array.

Decompose the array into P+1 equal parts.

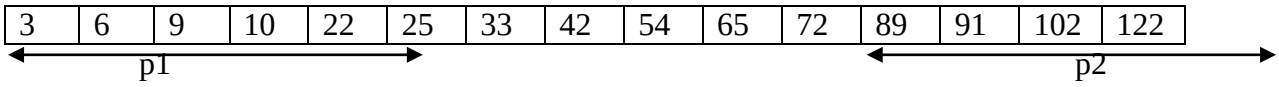
Then a processor check if y < x_k the last element in the sub array.

If y<$x_k$the right sub arrays are ignored else the left sub array is ignored. Anyway at one point we choose the left sub array or the right sub array.

Thus we keep on reducing the elements for search since it is sorted. Finally we arrive at P from where we can directly search p.

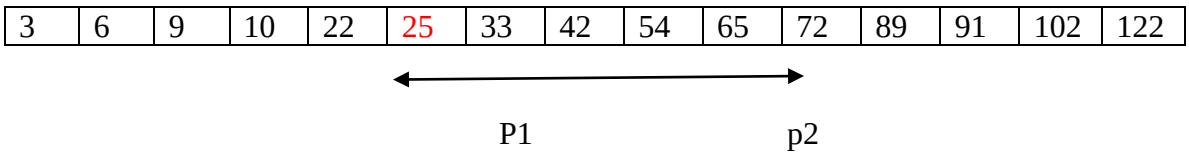
The complexity of parallel searching algorithm is O(log(n+1)/log(p+1))

Example:-



Let us take P=2 and y=25

Since n=15 we decompose the array in 3 subarrays.



Thus the element 25 is located by p1.

Animation

Note: Can be viewed only on Acrobat 9.0 and above.