J. Inform. Process. Cybernet. **EIK 30** (1994) 1, 29-43
(formerly: Elektron. Inform.verarb. Kybernet.)

Segmented Winner Trees[1]

By Andrew Lim and Sartaj Sahni

Abstract: A new data structure, *segmented winner tree*, is introduced. This is useful when one needs to represent data that are partitioned into segments. The segment operations that are efficiently supported are: initialize a unit length segment, find the element with least value in any given segment, update an element in any segment, merge two adjacent segments, and split a segment.
Key Words and Phrases: Data structures, winner trees, segmented data

1. Introduction

Suppose that we have n elements $1, 2, \ldots, n$ that have been partitioned into some number of segments such that the elements in each segment are contiguous. Figure 1 shows a possible partitioning of the eleven elements $1, 2, \ldots, 11$ into three segments. A value, $v[i]$, is associated with each element i.

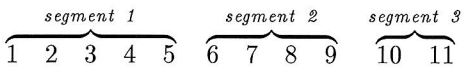

Figure 1: Eleven elements partitioned into three segments

The operations we wish to perform on these segments are:

1. *Initialize*(l_1): Initialize the one element segment $l_1 : r_1$ where $r_1 = l_1$.
2. *FindMin*(l_1, r_1, x): Find the smallest value in the segment comprised of elements $l_1 : r_1$. This value is returned in variable x.
3. *Update*(l_1, r_1, i, y): Change the value of element i to y. Element i is a member of the segment that is comprised of elements $l_1 : r_1$. Note that $l_1 \leq i \leq r_1$.
4. *Merge*(l_1, r_1, r_2): Merge the adjacent segments $l_1 : r_1$ and $r_1 + 1 : r_2$ into a single segment $l_1 : r_2$.
5. *Add*(l_1, r_1, y): This is a special case of merge in which the right segment $r_1 + 1 : r_2$ consists of a single element with value y. I.e., $r_1 + 1 = r_2$ and $v[r_2] = y$.

[1]This research was supported, in part, by the National Science Foundation under grant MIP91-03379.