Advanced Algorithms and Data Structures

Spring 2017

Wolfgang Jeltsch
Tiina Zingel

Homework 6

Submission on 24 April 2017

Task 1 (Implementation of UNION–FIND) *6 points*

The course website contains a partial implementation of partitions. The implementations of the function *Part* and the internal procedure *Link* are missing. Add implementation of these subprograms that use the UNION–FIND approach with path compression.

Please note the following about the package *ITI8590.Partitions*:

- The package supports only partitions of integer intervals. This allows for an efficient implementation of the forests that represent partitions. We implement such a forest as an array that is indexed by the elements of the base set. The element at an index i is the node that corresponds to i. To reference the parent of a node, we do not store a pointer to the parent, but the value of the parent. As a result, we cannot use a null pointer to signal that a node is a root. Instead we let the parent reference of a root node point to the root node itself, that is, we store the value of the root node itself in the *Parent* field of the root node.

- We dynamically allocate the arrays that represent partitions, and reference them via pointers. That way, the amount of memory they use can be determined at runtime.

- The subprogram *Part* is supposed to realize the FIND operation. Since *Part* is a function, it must have only **in** parameters. This seems to be okay, since the FIND operation does not change the given partition and the element to look up. The problem is that FIND must perform path compression and thus change the internal representation of the partition. However the *Part* function can actually change the array that represents the partition, since this array is not passed directly as a parameter, but referenced by a pointer. Only this pointer cannot be changed.