Data Structures and Algorithms
Algorithms and Complexity
Exam
9:00 - 12:00, June 22, 2005

Prof. Dr. M. H. Böhlen

The exam consists of 6 exercises. For each exercise you can get 30 points. It is important that you argue for your answers and that you present your solutions in a readable form. Fully specify all algorithms that you use as part of your solutions.

Exercise 1

The core of quicksort is the partitioning algorithm. Below are two specific partitioning algorithms: P1 is the partitioning proposed by Lomuto and P2 is the partitioning proposed by Hoare.

P1(A,l,r)

```
x := A[r]; i := l-1;
FOR j := l TO r-1 DO
  IF A[j] ≤ x THEN i++; swap A[i] and A[j];
swap A[i+1] and A[r];
RETURN i+1;
```

P2(A,l,r)

```
x := A[l]; i := l-1; j := r+1;
WHILE TRUE DO
  REPEAT j-- UNTIL A[j] ≤ x;
  REPEAT i++ UNTIL A[i] ≥ x;
  IF i < j THEN swap A[i] and A[j]; ELSE RETURN j;
```

Quicksort calls the partitioning procedure repetitively and each call of the partitioning algorithm partitions the sub-array A[l..r].

1. Describe the ideas of the two partitioning strategies. For each strategy illustrate the state at the beginning, in the middle, and at the end.

2. Give appropriate invariants for both partitioning algorithms.

3. Explain what happens if in line 03 $\leq$ is replaced by $<$.
4. Explain what happens if in line 11 A[l] is exchanged with A[r].
5. For each partitioning algorithm give an appropriate quicksort algorithm.

Exercise 2

The *edit distance* between source string x and target string y is the cost of the least expensive sequence of operations that transforms string x into string y. Permitted operations to transform x into y are

- to copy a character from x to y,
- to replace a character from x by another character,
- to delete a character from x, or
- to insert a character into y.

For instance with $x =$ "house" and $y =$ "home" a possible sequence S of edit operations that transforms x into y is:

$$S = \text{copy h, copy o, replace u by m, delete s, copy e}$$

We assume operations with the following unit costs: copy (2), replace (5), insert (4), and delete (3). With these costs the sequence S has cost 14. Since there is no sequence that transforms x into y and is cheaper the edit distance between x and y is 14.

1. Develop a recursive algorithms that computes the edit distance between a source string x and a target string y.
2. Use dynamic programming to develop an efficient algorithm to compute the edit distance between two strings. Discuss the data structure you use for your solution and illustrate it for the strings $x =$ "house" and $y =$ "home".

Exercise 3

Assume a set of countries. Each country corresponds to an administrative unit. Countries can be sub-divided into a hierarchy of smaller administrative units. For instance Italy is subdivided into provinces, which are sub-divided into regions, which are divided into areas, municipalities, etc.
A *geographic region* is an administrative unit. We say that a geographic region r is divided into a number of sub-regions and each of these subregions

has r as their father region. For example South Tyrol is the father region of Sarntal, and Sarntal is a sub-region of South Tyrol.
A geographic region is represented by an instance of the GeoRegion abstract data type with the following operations.

area(r: GeoRegion): int
: $area(r)$ returns the area of the geographic region r measured in square kilometers.

firstSubregion(r: GeoRegion): GeoRegion
: $firstSubregion(r)$ returns the first region in the list of sub-regions of the geographic region r. It returns NULL if region r is not subdivided into smaller regions.

siblingSubregion(r: GeoRegion): GeoRegion
: $siblingSubregion(r)$ returns the next region (in an arbitrary order) in the list of sub-regions that has the same father region as r. For example, Ultental could be a sibling sub-region of Sarntal. It returns NULL if all subregions have been returned or if r is not part of any larger region.

removeSubregion(r, s: GeoRegion)
: $removeSubregion(r, s)$ removes the geographic region s from the list of subregions of r (if s in that list).

addSubregion(r, s: GeoRegion)
: $addSubregion(r, s)$ adds the geographic region s to the end of the list of subregions of r. If s was a subregion of another region r' the operation first removes s from the list of subregions of r'.

Use the GeoRegion abstract data type to solve the following problems:

1. Write an algorithm that counts how many direct large subregions a geographic region r has. A region is called large if its area is more than 100 km^2.

2. The area of a geographic region is *consistent* if its area is equal to the sum of the areas of its sub-regions and the areas of its subregions are consistent. If a region does not have subregions its area is consistent. Write an algorithm that checks if the area of a geographic region is consistent.

Exercise 4

Consider the Red-Black tree in Figure 1. Note that instead of colors we use circles for black nodes and rectangles for red nodes. Thus, 2, 5, 6, 7, and 9 are black nodes, whereas 3, 4, 8, and 11 are red nodes.

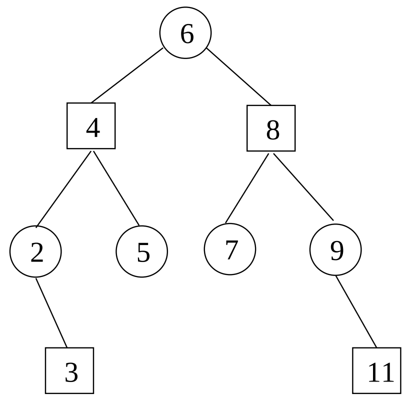

Figure 1: Red-Black Tree

Perform the following operations on this tree:

1. insert 12
2. delete 5
3. delete 6
4. insert 10

Discuss the different cases you encounter and illustrate all steps of the tree reorganizations.

Exercise 5

Assume arithmetic expressions that consist of integers, the four arithmetic operations +, -, *, and /, and parentheses. An example is the expression:

$$(51 + 6) * 72 - (9 - 2)/3$$

1. Represent the above expression as a binary tree.
2. Give appropriate declarations in C to represent such a tree.
3. In the reverse Polish notation the operator follows the argument expressions. Thus, (5 + 16) * 3 would be specified as 5 16 + 3 *. Give a C algorithm that starts with a binary tree that represents an arithmetic expression, and prints this expression in reverse Polish notation.
4. Write a C function that evaluates an arithmetic expression (represented as a tree) and returns the result. You can assume integer operations throughout

Exercise 6

Paul Erdös (1913-1996) was a prolific Hungarian mathematician who wrote hundreds of mathematical papers in many different areas and with various coauthors. Each scientist, knowingly or unknowingly, has an associated Erdös number that is defined through the following procedure:

- Paul Erdös has Erdös number 0.
- A person not yet assigned an Erdös number who has a joint publication with a person having Erdös number n has Erdös number $n + 1$.
- Persons who do not get assigned an Erdös number by the above process have Erdös number ∞.

For example Albert Einstein has Erdös number 2 because he did not have a joint publication with Erdös but wrote two joint papers with his Princeton assistant Ernst Straus with whom Erdös wrote 20 papers.

1. What model would you use to describe the collection of all scientists and their research connections? For our purposes a research connection is established if two people write a joint paper.
2. Based upon your model give a general algorithm that, given a scientist X, returns X's Erdös number. What is the complexity of your algorithm?
3. According to the above numering system each scientist who has written a joint paper with Erdös has Erdös number 1. A refined approach weighs the collaboration between two scientists and assigns the weight $1/n$ if n is the number of their joint papers. Thus, according to this refined numbering scheme the collaboration between Erdös and Straus has weight $1/20$. The weight of a sequence of collaborations is the sum of its components. Thus, the weight of the sequence Erdös-Straus-Einstein is $11/20$. Assume we want to determine all scientists that can be reached via sequences of collaborations that start with Erdös and have a weight smaller than 1. Give a general algorithm for this problem and analyze its complexity.