Technische Universität München
Institut für Informatik
Scientific Computing in Computer Science

WS 2010/11
Worksheet
30.11.2010

Fundamental Algorithms 4

Exercise 1

Consider the algorithm

```
CountingSort(A: Array[1..n], N:Integer) {
   for i from 1 to n do {
      C[A[i]] := C[A[i]] + 1
   }
   for i from 2 to N do {
      C[i] := C[i] + C[i-1]
   }
   for i from n downto 1 to {
      B[C[A[i]]] := A[i];
      C[A[i]] := C[A[i]] - 1
   }
   for i from 1 to n do {
      A[i] := B[i]
   }
}.
```

`CountingSort` will sort the array `A`, if all array elements are between 1 and N.

(a) for the input array `[6,3,7,5,4,3,3,2,8,9,1,6]`, and $N = 9$, specify the contents of the arrays `B` and `C`

- after the first for-loop;
- after the second for-loop;
- after each of the first three iterations of the third loop;
- after finishing the third loop.

(b) compute the number of arithmetic operations of `CountingSort` depending on the values of n and N.

(c) in the third for-loop, replace the statement '`for i from n downto 1 do`' by '`for i from 1 to n do`'. Show that the algorithm is still correct. What is the difference between the two versions?

Solution:

(a) After the first for-loop, the array `C` will contain the number of occurences of each value in the array `[6,3,7,5,4,3,3,2,8,9,1,6]`:

`C = [1,1,3,1,1,2,1,1,1]`

After the second for-loop, each element `C[i]` will contain the number of elements in `A` that are smaller or equal to `i`:

`C = [1,2,5,6,7,9,10,11,12]`

The third loop will copy the elements from `A` into their correct positions in array `B`. The contents of `B` and `C` will be

- after the first iteration:
  `B = [*,*,*,*,*,*,*,*,6,*,*,*], C = [1,2,5,6,7,8,10,11,12]`
- after the second iteration:
  `B = [1,*,*,*,*,*,*,*,6,*,*,*], C = [0,2,5,6,7,8,10,11,12]`
- after the third iteration:
  `B = [1,*,*,*,*,*,*,*,6,*,*,9], C = [0,2,5,6,7,8,10,11,11]`

The `*` indicates that the respective value will depened on the initialization of the array `B` (the initial values of the array `B` are not explicitely definied in the algorithm). After the third for-loop has finished, the contents of `B` will be the sorted array: `B = [1,2,3,3,3,4,5,6,6,7,8,9]`

(b) There are three loops over all elements (n iterations) plus one loop of the possible element values ($N-1$ iterations). Hence, the number of artihmetic operations is $\Theta(N) + \Theta(n) = \Theta(N+n)$.

(c) The final array will be sorted regardless of the sequence of loop executions. However, if the loop is executed as given on the worksheet, the elements retain their relative order – i.e., elements with the same values will stay in the same relative order. For a loop '`for i from 1 to n do`', their relative order will be exactly inverted.