

Fundamental Algorithms

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 $[0, 3, 7, 5, 4, 3, 3, 2, 8, 9, 1, 0]$, 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 (see lecture).
- (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?

Exercise 2

Prove the following statement:

If $c \cdot n$ comparisons are sufficient to compute the median of a set of n elements, then a variant of quicksort that chooses the median as pivot element will require at most $(c + 1)n \log n + O(n)$ comparisons.