

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 $[6, 3, 7, 5, 4, 3, 3, 2, 8, 9, 1, 6]$, 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 (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?

Solution:

- (a) After the first for-loop, the array C will contain the number of occurrences 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:

$$B = [*, *, *, *, *, *, *, *, 6, *, *, *], \quad C = [1, 2, 5, 6, 7, 8, 10, 11, 12]$$

- after the second iteration:

$$B = [*, *, *, *, 3, *, *, *, 6, *, *, *], \quad C = [1, 2, 4, 6, 7, 8, 10, 11, 12]$$

- after the third iteration:

$$B = [*, *, *, *, 3, *, *, *, 6, 7, *, *], \quad C = [1, 2, 4, 6, 7, 8, 9, 11, 12]$$

The $*$ indicates that the respective value will depend on the initialization of the array B (the initial values of the array B are not explicitly defined 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) see lecture

- (c) t.b.c.

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.

Solution:

Assume that the partitioning requires n comparisons, the total number of comparison of this median-quicksort will be

$$\begin{aligned}C(1) &= 0 \\C(n) &= c \cdot n + 2C\left(\frac{n}{2}\right) + n = (c+1)n + 2C\left(\frac{n}{2}\right) \\&= (c+1)n + 2\left((c+1)\frac{n}{2} + 2C\left(\frac{n}{4}\right)\right) = (c+1)n + (c+1)n + 4C\left(\frac{n}{4}\right) \\&= \dots \\&= k \cdot (c+1)n + 2^k C\left(\frac{n}{2^k}\right)\end{aligned}$$

If $k = \lceil \log_2 n \rceil$, then $C\left(\frac{n}{2^{\lceil \log_2 n \rceil}}\right) = C(1) = 0$, and therefore:

$$C(n) = (c+1)nk = (c+1)n \lceil \log_2 n \rceil$$