Department of Informatics
Database Technology Group
University of Zurich Prof. Dr. Anton Dignös

Informatics II
Exercise 3

Mar 08, 2020

Goals:

- practice calculation of asymptotic tight bound.
- practice running time analysis
- practice best case and worst case analysis
- identify the influence of a parameter

Algorithmic Complexity and Correctness

Task 1. Consider algorithm `whatDoIDo(A, n, k)` below. Input array `A[]` contains n integers, and k is an integer.

```
Algo: WHATIDO(A, n, k)
sum = 0;
for i = 1 to k do
    maxi = i;
    for j = i + 1 to n do
        if A[j] > A[maxi] then
            maxi = j;
    sum = sum + A[maxi];
    swp = A[i];
    A[i] = A[maxi];
    A[maxi] = swp;
return sum
```

a) What does algorithm `whatDoIDo(A,k)` do?

b) Implement algorithm `whatDoIDo(A,k)` in C, and call your implementation in a complete C program.

c) Conduct an exact analysis of the running time of algorithm `whatDoIDo(A,k)`.

d) Determine the best and the worst case of the algorithm. What is the running time and asymptotic complexity in each case?

e) What influence does the parameter k have in the asymptotic complexity?

f) List special cases, and provide an example for each special case if possible.

Task 2. Calculate the asymptotic tight bound for the following functions and rank them by their order of growth (lowest first). Clearly work out the calculation step by step in your solution.

$$\begin{aligned}
f_1(n) &= (n+3)! \\
f_2(n) &= 2\log(6^{\log n^2}) + \log(\pi n^2) + n^3 \\
f_3(n) &= 4^{log_2 n} \\
f_4(n) &= 12\sqrt{n} + 10^{223} + \log 5^n \\
f_5(n) &= 10^{\lg 20} n^4 + 8^{229} n^3 + 20^{231} n^2 + 128 n \log n \\
f_6(n) &= \log n^{2n+1} \\
f_7(n) &= 101^{\sqrt{n}} \\
f_8(n) &= \log^2(n) + 50\sqrt{n} + \log(n) \\
f_9(n) &= n^n + 2^{2n} + 13^{124} \\
f_{10}(n) &= 14400
\end{aligned}$$

Task 3. Let n be an exact power of 2, $n = 2^k$.
Use mathematical induction over k to show that the solution of the recurrence involving positive constants $c, d > 0$

$$T(n) = \left\{ \begin{array}{ll} d, & \text{if } n = 2^0 = 1 \\ 2T(n/2) + cn & \text{if } n = 2^k \text{and } k \geq 1 \end{array} \right\}$$

is $T(n) = dn + cn\ log(n)$

Hint: you may want to rewrite the above as $T(2^k) = d2^k + c2^k log(2^k) = d2^k + c2^k \cdot k$.