Barcelona - Nice - Hamburg - L'Aquila - Gdansk

Master Mathmods, march 4, 2012

Complexity and recursivity

Order of magnitude

Prove the following:

(1) If $\alpha, \beta \geq 1$ are two real-valued constants, then $\log_\alpha(n) = \Theta(\log_\beta(n))$

(2) $u = v + O(f)$ iff $v = u + O(f)$

Iterative complexity

Estimate the complexity of the following programs. We assume instructions `I1, I2, I3` in $O(1)$.

```
for i in 1..n-1
   for j in i+1..n
      for k in i..j
         I1
      end
   end
end
```

```
i,j = 1,1
while i < n
   i+=1
   I1
   while j < n and Condition
      j+=1
      I2
   end
   I3
end
```

Recursive complexity

We consider the next two algorithms' complexity for computing the product of two large integers. Let U and V two integers of size $2n$ in basis β such that $U = A\beta^n + B$ and $V = C\beta^n + D$.

- We first use the equality $(A\beta + B)(C\beta + D) = A.C\beta + \beta(A.D + B.C) + B.D$. Instead of multiplying two integers of size $2n$, we have to compute four products of integers of size n, three shifts (multiplication by β) and three additions. We assume that the complexity of the last two arithmetic operations is $O(n)$ and that $T(1) = k$. First write the recurrence relation between $T(2.n)$, the time complexity for multiplying two integers of size n and $T(n)$. Solve this recurrence relation.

- Same question when using the equality $(A\beta+B)(C\beta+D) = A.C\beta 2+((A-B).(D-C)+A.C+B.D)\beta + B.D$. First, count the number of basic arithmetical operations and assume also $T(1) = k$. The latter equality is known as Toom-Cook multiplication

Limits of the recurrence-solving theorem

Show that the theorem for solving recurrence relations does not apply to $T(n) = 2T(n/2) + n.\log(n)$. Do you know any other means for solving this recurrence?