What is...the else function?

Or: Counting else

Recursion

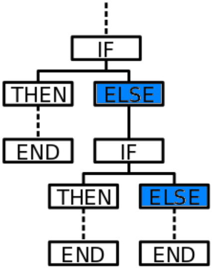
- Recursion = something that calls itself
- Example $f(0) = 1$ and $f(n) = n \cdot f(n-1)$ for $n > 0$

Pseudocode (recursive):

```
function factorial is:

input: integer n such that n >= 0

output: [n × (n-1) × (n-2) × ... × 1]


    1. if n is 0, return 1
    2. otherwise, return [ n × factorial(n-1) ]


end factorial
```

- Recursion is often very efficient

Else



- Else function :

Algorithm The Else function $Else(x, y, z)$:

if $x \leq y$ **then**
  return y
else
  return $Else(Else(x-1, y, z), Else(y-1, z, x), Else(z-1, x, y))$
end if

- The function essentially consists of else only

Recursion only...?

[Takeuchi 1978] Ikuo Takeuchi. On a Recursive Function That Does Almost Recursion Only. Memorandum, Musahino Electrical Communication Laboratory, Nippon Telephone and Telegraph Co., Tokyo, 1978.

A000651 Running time of Takeuchi function. +30 2

0, 1, 4, 14, 53, 223, 1034, 5221, 28437, 165859, 1029803, 6772850, 46983238, 342509396, 2615606677, 20865444825, 173446634597, 1499111445237, 13445550920288, 124919896067530, 1200320663197275, 11910845573790488 (list; graph; refs; listen; history; text; internal format)

OFFSET 0,3

REFERENCES
D. E. Knuth, personal communication.
V. Lifschitz, editor, Artificial intelligence and mathematical theory of computation. Papers in honor of John McCarthy. Academic Press, Inc., Boston, MA, 1991. See p. 215.
T. Prellberg, On the asymptotics of Takeuchi numbers, Symbolic computation, number theory, special functions, physics and combinatorics, Kluwer Acad. Publ., Dordrecht, 2001, pp. 231-242. MR 2002m:11016.

- **Count else** Let $T(x, y, z)$ = number of times the else clause is invoked

$$T_n = T(n, 0, n+1)$$

- **Example** $T_2 = 4$:

$$
\begin{aligned}
Else(2,0,3) &= Else(Else(1,0,3), Else(-1,3,2), Else(2,2,1)) && \text{1 else}\\
&= Else(Else(1,0,3),3,2)\\
Else(1,0,3) &= Else(Else(0,0,3), Else(-1,3,1), Else(2,1,0)) && \text{1 else}\\
&= Else(0,3,Else(2,1,0))\\
Else(2,1,0) &= Else(Else(1,1,0), Else(0,0,2), Else(-1,2,1)) && \text{1 else}\\
&= Else(1,0,2)\\
Else(1,0,2) &= Else(Else(0,0,2), Else(-1,2,1), Else(1,1,0)) && \text{1 else}\\
&= Else(0,2,1) = 2
\end{aligned}
$$

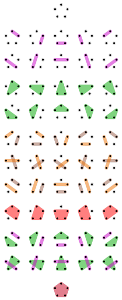
Enter, the theorem

We get the following asymptotic formula:

$$T_n \sim C \cdot B_n \cdot \exp(1/2 \cdot W(n)^2)$$

- $B(n)$ = Bell numbers = number of partitions of a set of size n Grow fast

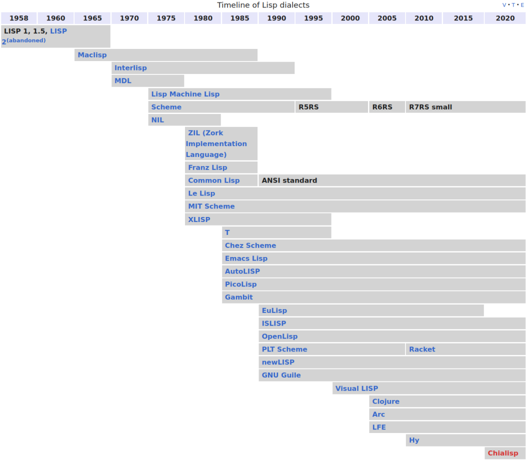
$B(n) \gg \exp(n)$:



- $W(n)$ = Lambert's W function (grows roughly as $\log(n)$ – ignore)
- $C \approx 2.2394331040...$

Comparing computation speeds

timeline of LISP : dialects



- Else was developed ∼1978 to compare the speeds of LISP systems
- The point It can run a long time without creating large numbers etc.
- Later ∼1991 it was then studied as a sequence

Thank you for your attention!

I hope that was of some help.