Numărați tipurile

Definim recursiv: un *tip* este fie tipul de bază T fie un tip de funcție $T_1 \to T_2$, unde T_1 și T_2 sunt tipuri.
Asociem fiecărui tip un număr de ordine unic: $\mathsf{N}(\mathsf{T}) = 0$, $\mathsf{N}(T_1 \to T_2) = 2^{\mathsf{N}(T_1)} \cdot (2 \cdot \mathsf{N}(T_2) + 1)$.
Astfel, tipurile în ordine crescătoare a numerelor sunt: T, $\mathsf{T} \to \mathsf{T}$, $(\mathsf{T} \to \mathsf{T}) \to \mathsf{T}$, $\mathsf{T} \to (\mathsf{T} \to \mathsf{T})$, ...
Când T_1 sau T_2 sunt tipuri de funcție, se scriu între paranteze.

Programul va citi din fișierul `tipuri.in` un număr natural `n` .
Programul va scrie în fișierul `tipuri.out` pe două linii:
primul tip care se scrie cu `n` săgeți și numărul său de ordine
(presupus a încape în cel mai mare întreg admis de limbaj).
Tipurile se scriu fără spații, și cu `->` pentru săgeată.

Exemplu

`tipuri.in`	`tipuri.out`
`3`	`((T->T)->T)->T`
	`4`

Soluție Pentru $n > 0$, tipul căutat are forma $T_1 \to T_2$, unde T_1 și T_2 au n_1, respectiv n_2 săgeți și $n_1+n_2 = n-1$. Cum $2^{N(T_1)}(2\,N(T_2)+1)$ e crescătoare în $N(T_1)$ și $N(T_2)$, T_1 și T_2 trebuie să aibe și ele număr de ordin minim. Dacă M_n e ordinul minim pentru tipuri cu n săgeți, avem: $M_0 = 0$, $M_n = \min\limits_{n_1+n_2=n-1} 2^{M_{n_1}}(2\,M_{n_2}+1)$.

Avem deci o problemă obișnuită de programare dinamică: putem calcula M_n și stoca valorile într-un tabel: $M_1 = 2^0(2 \cdot 2^0 + 1) = 1$, $M_2 = \min\{2^{M_0}(2\,M_+1), 2^{M_1}(2\,M_0+1)\} = 2$, $M_3 = 4$, $M_4 = 9$, etc.

Calculul direct al minimului pentru toate combinațiile $n_1 + n_2 = n - 1$ nu e fezabil pentru că M_n crește exponențial (comparabil cu 2^{n-1}), iar unele variante din calculul minimului chiar dublu exponential: încercând $n_1 = n-1, n_2 = 0$, obținem $2^{M_{n-1}} \approx 2^{2^{n-2}}$ care depășește rapid ($n = 8$) domeniul întregilor reprezentabili. Cum 2^k crește mult mai repede decât $2k+1$, pentru minim trebuie să încercăm valori mici pentru n_1:
$n_1 = 0, M_0 = 0, n_2 = n-1 \to 1(2\,M_{n-1}+1)$ Exemplu (n=5): 19
$n_1 = 1, M_1 = 1, n_2 = n-2 \to 2(2\,M_{n-2}+1)$ Exemplu (n=5): 18
$n_1 = 2, M_2 = 2, n_2 = n-3 \to 4(2\,M_{n-3}+1)$ Exemplu (n=5): 20
Pentru $n_1 = 3$ avem $M_3 = 4$ și obținem $16(2\,M_{n-4}+1)$, care, presupunând $M_n \approx 2^{n-1}$ e deja de circa două ori mai mare decât variantele anterioare (exemplu pentru n=5: 48).

Observația simplifică obținerea minimului, din mai puține variante, și rezolvă problema depășirii premature, dar trebuie să fim siguri că e corectă, ca să nu pierdem un caz valid de minim. Putem face acest lucru simplu, calculând minimul m din primele 3 cazuri ($n_1 = 0, 1, 2$), și verificând pentru restul cazurilor logaritmul: $\log(2^{M_{n_1}}(2\,M_{n_2}+1)) > \log m$, adică $M_{n_1} + \log(2\,M_{n_2}+1) > \log m$, ceea ce se poate face fără depășire.

Scrierea unui tip $T(N)$ știind numărul de ordin N se face simplu recursiv: $T(0) = \mathsf{T}$, iar $N > 0$ se descompune în forma $2^{N_1}(2\,N_2+1)$, și scriem $T(N) = T(N_1) \to T(N_2)$, punând paranteze în jurul tipurilor compuse.

Analizând mai atent, putem demonstra prin inducție o recurență în funcție de restul modulo 3 al lui n: $M_0 = 0, M_{3k+1} = 2\,M_{3k}+1, M_{3k+2} = 2\,(2\,M_{3k}+1), M_{3k+3} = 4\,(2\,M_{3k}+1)$, iar pentru tipul de ordin minim: $T_0 = \mathsf{T}$, $T_{3k+1} = \mathsf{T} \to T_{3k}$, $T_{3k+2} = (\mathsf{T} \to \mathsf{T}) \to T_{3k}$, $T_{3k+3} = ((\mathsf{T} \to \mathsf{T}) \to \mathsf{T}) \to T_{3k}$, ($k \geq 0$), deci putem calcula răspunsul din 3 în 3. (Demonstrați că $2^{n-1} \leq M_n \leq 2^n - 1$ pentru $n > 0$, și că $n_1 > 2$ nu poate da minimul, apoi arătați că din cazurile $n_1 = 0, 1, 2$, minimul corespunde relațiilor date).

Ca exercițiu matematic suplimentar: demonstrați că $M_n = \lfloor 2^{n+2}/7 \rfloor$.

Testele de concurs:

```
 0: 0          T
 4: 9          T->(((T->T)->T)->T)
 8: 146        (T->T)->(((T->T)->T)->(((T->T)->T)->T))
10: 585        T->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->T)))
14: 9362       (T->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->T))))
19: 299593     T->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)
               ->(((T->T)->T)->T))))))
24: 9586980    ((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)
               ->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->T)))))))
26: 38347922   (T->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)
               ->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->T))))))))
29: 306783378  (T->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)
               ->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->T)))))))))
31: 1227133513 T->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)
               ->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->(((T->T)->T)->T))))))))))
```