
Clojure — La Recurrència amb Trampolí**W83336_ca**

Es pot demostrar que la recurrència $u_{n+2} = 5 \times u_{n+1} - 6 \times u_n$, amb $u_0 = 0$ i $u_1 = 1$ és tal que per a tot n , $u_n = 3^n - 2^n$.

L'estudiant que no ho sap, però vol conèixer el valor d' u_n per a qualsevol n , podria ser tan ingenu d'implementar la següent funció d'una manera molt ineficient, fent servir la definició matemàtica:

```
(defn recurrència [n]
  (cond
    (or (= n 0) (= n 1)) n
    :else (let [rm1 (recurrència (dec n))
                rm2 (recurrència (dec (dec n)))]
              (- (* 5 rm1) (* 6 rm2)))))
```

Passeu aquesta funció a *continuation passing style* per obtenir-ne la versió recursiva final. Anomeneu-la **recurrència-cps**.

Després, *trampolinitzeu* `recurrència-cps` per evitar d'obtenir l'error de *Stack Overflow*. Anomeneu-la **recurrència-t**.

Observacions

Cal que envieu un fitxer amb la definició de les **dues** funcions `recurrència-cps` i `recurrència-t`.

Exemple d'entrada 1

```
(recurrència-cps 5 identity)
(recurrència-cps 10 identity)
(recurrència-cps 15 identity)
(recurrència-t 5)
(recurrència-t 10)
(recurrència-t 15)
(recurrència-t 20)
```

Exemple de sortida 1

```
211
58025
14316139
211
58025
14316139
3485735825
```

Informació del problema

Autoria: Jordi Delgado

Generació: 2026-01-25T13:20:35.472Z