
Clojure — Parèntesis amb Backtracking

S28790_ca

Implementeu una funcions *solucions* que ens doni totes les permutacions correctes de n parelles de parèntesis tot seguint l'esquema *backtracking* descrit als apunts de l'assignatura.

Recordeu afegir al vostre codi solució el següent:

```
(defn bTck [succ obj] ;; ja coneixeu flip i foldr, les fem servir aquí
  (letfn [(bTck' [v]
            (cond
              (empty? v) []
              (obj (peek v)) (conj (bTck' (pop v)) (peek v))
              :else (let [x (peek v)]
                      (recur (foldr (flip conj) (pop v) (succ x))))))]
    (fn [inicial]
      (bTck' [inicial])))
```

Exemple d'entrada 1

```
(solucions 3)
```

Exemple de sortida 1

```
((()()) ()()) (())() (())() (()))
```

Informació del problema

Autoria: Gerard Escudero

Generació: 2026-01-25T12:43:35.146Z

© Jutge.org, 2006–2026.

<https://jutge.org>