

テキスト処理 第9回 (2006-06-20)

木構造探索レポート説明

田中哲

産業技術総合研究所

情報技術研究部門

`akr@isc.senshu-u.ac.jp`

`http://staff.aist.go.jp/tanaka-akira/textprocess/`

レポート

- ネストしているかもしれない配列 obj と整数 val が与えられたときに、obj 内に val が含まれているときに true, 含まれていなければ false を返すメソッドを定義せよ
- `def search(obj, val) ... end`
- ユニットテストを提供するので、実装したらテストして確認すること
- ✕切 2006-06-20 16:20
- IT's class
- 拡張子が txt なテキストファイルを望む

実装1

eachメソッドがあったら
(配列だったら)

```
• def search(obj, val)
  if obj.respond_to? :each
    result = false
    obj.each { |v| result ||= search(v, val) }
  else
    val == obj
  end
end
```

それぞれで見つかったか
どうかを論理和

末端だったら比較
同じだったら true
違ったら false

実装2: 見つけたところで中断

- ```
def search(obj, val)
 if obj.respond_to? :each
 obj.each { |v| return true if search(v, val) }
 false
 else
 val == obj
 end
end
```
- ブロックつき呼び出しの中断は今日説明する

# ざっと見た結果

- 過半数はちゃんとできていた気がする
- いくつかの原因
  - 再帰をうまく作れない
  - 比較対象が配列かもしれないと考えてしまったかも？
  - 表示することと値を返すことの違い
- flatten を使っている人がいた (ううむ)

# 再帰の作り方

1. とりあえず再帰を使わずに書きかけてみる
2. 無限に書くはめになったら再帰が必要
3. 同じ問題が繰り返し出てくる場所を見つけてそこを再帰でまとめる

# search: 問題

- def search(obj, val)  
 obj の中に val が入っているか?  
end

# obj が配列かどうか場合分け

- def search(obj, val)  
 if obj.respond\_to? :each # obj は配列  
 配列 obj の中に val が入っているか?  
 else # obj は配列でない  
 配列でない obj は val か?  
 end  
end



# 配列でなければ比較しておしまい

- ```
def search(obj, val)
  if obj.respond_to? :each # obj は配列
    配列 obj の中に val が入っているか?
  else # obj は配列でない
    val == obj
  end
end
```

配列のときは要素ごとに処理

- ```
def search(obj, val)
 if obj.respond_to? :each
 result = false
 obj.each { |v| result ||= (vの中にvalが入っているか?) }
 result
 else
 val == obj
 end
end
```

# v が配列かどうか場合分け

- ```
def search(obj, val)
  if obj.respond_to? :each
    result = false
    obj.each {|v| result ||= (if v.respond_to? :each
                              配列 v の中に val が入っているか?
                              else
                              配列でない v は obj か?
                              end)
    }

    result
  else
    val == obj
  end
end
```

相似な構造が見つかった

- def search(obj, val)

```
if obj.respond_to? :each
```

```
  result = false
```

```
  obj.each { |v| result ||= (if v.respond_to? :each
```

```
    配列 v の中に val が入っているか?
```

```
    else
```

```
    配列でない v は obj か?
```

```
    end)
```

```
  }
```

```
  result
```

```
else
```

```
  val == obj
```

```
end
```

```
end
```

相似な所は外側への呼び出しで済みます

- def search(obj, val)

```
if obj.respond_to? :each
  result = false
  obj.each { |v| result ||= search(v, val) }
  result
else
  val == obj
end
```

```
end
```