



本日の課題 (1)



Gnome (ノーム) ソートアルゴリズムについて、以下のWikipediaの記事を参考にして、JavaScriptでこのソートアルゴリズムを実現せよ。具体的には関数 `kadaai(lst)` を定義せよ。lstは数のリストであり、関数の返り値はlstを小さい方から順番に大きくなるように並べたリストになるようにせよ。

<http://ja.wikipedia.org/wiki/%E3%83%8E%E3%83%BC%E3%83%A0%E3%82%BD%E3%83%BC%E3%83%88>

この関数を用いた場合、almost sortedなデータ（ほとんどソートされているが、一部順番が崩れているようなデータ）について高速にソートされることを確認せよ。almost sortedな数のリストを生成するプログラムはつぎのスライドに示す。

本日の課題 (2)

almost sortedな数のリストを生成する関数

```
function almost_sorted(n){
  var lst = []
  var N = Math.floor(n * 0.8)
  for (var i = 0; i < n; i++){
    lst.push(Math.random())
  }
  lst.sort(function (x, y){
    if (x < y) return -1
    else if (x > y) return 1
    else return 0
  } )
  for (var k = 0; k < N; k++){
    var i = Math.floor(Math.random() * (n - 5))
    var d = Math.floor(Math.random() * 5)
    var temp = lst[i]
    lst[i] = lst[i + d]
    lst[i + d] = temp
  }
  return lst
}
```

```
almost_sorted(20) =
[0.3308449019904972,0.30249652232490454,0.120568362023986
57,0.11172801615123418,0.2039010779657161,0.3656404764192
5446,0.42568469103496687,0.4968298584874332,0.48538740455
44003,0.5653550135696107,0.5730692039281771,0.65643420109
33445,0.6806016400656708,0.7906112778012458,0.74611004556
69576,0.8206033959039577,0.8264918353558333,0.83058243716
13544,0.8944503899665174,0.9060165336873575]
```