## 練習問題 2

クイックソートでのソーティングが最悪ケースに陥ってしまう場合を考える．クイックソートの実現方法は一通りではないが，ここでは講義内容に合わせ，注目領域の左端の値を分割操作の基準値とする．すなわち，左端の値よりも大きいか小さいかという視点でデータの整理と分割を行っていく．

$n$ 個の値 $a_1, a_2, \ldots, a_n$ が与えられ，これらを昇順（小さい順）に並べ替えるのに要する計算時間を $T(n)$ という $n$ の関数で表すことにする．そこで偶然にもこれらは既に昇順に並んでいたとする．それに気付かずにクイックソートを実行すると以下の流れになる．

(1) $a_1$ を基準として，$a_1$ より小さいものと大きいものに分ける．
$a_1$ とそれぞれ比較するので比較回数は $n-1$ 回となる．なお，全てが $a_1$ より大きいので値の交換は行われない．よって，この操作にかかる時間は $c(n-1)$ といえる（$c$ は定数）．

(2) 次に $a_1$ より小さいグループと大きいグループに対してそれぞれ再帰的にクイックソートを実行する．しかし (1) の結果，$a_1$ より小さいものは無いためそちらの処理は終了する．一方，$a_1$ より大きな値は $n-1$ 個ある．それゆえ，$a_1$ より大きな値に対する処理時間は $T(n-1)$ と書ける．

以上より，$T(n)$ は

$$T(n) = T(n-1) + c(n-1)$$

という漸化式で書ける．ただし，$n = 1$ の場合は何もしなくてよいので，ここでは $T(1) = c$ とする．

この漸化式を解き，計算量をビッグオー記法で答えなさい．

[答え] 次ページ参照.

## 解答例

まずは漸化式を解く．

$$
\begin{aligned}
T(n) &= T(n-1) + c(n-1) \\
&= \{\ T(\ (n-1) - 1\ ) + c(\ (n-1) - 1\ )\ \} + c(n-1) \\
&= T(n-2) + c(n-2) + c(n-1) \\
&= \{\ T(\ (n-2) - 1\ ) + c(\ (n-2) - 1\ )\ \} + c(n-2) + c(n-1) \\
&= T(n-3) + c(n-3) + c(n-2) + c(n-1) \\
&\vdots \\
&= T(n-k) + c(n-k) + c(n-k+1) + \cdots + c(n-2) + c(n-1) \\
&= T(n-k) + c\sum_{i=1}^{k}(n-i)
\end{aligned}
$$

$k = n-1$ まで展開していくと

$$
\begin{aligned}
&= T(\ n-(n-1)\ ) + c\sum_{i=1}^{n-1}(n-i) \\
&= T(1) + c\sum_{i=1}^{n-1}(n-i)
\end{aligned}
$$

$T(1) = c$ なので

$$
\begin{aligned}
&= c + c\sum_{i=1}^{n-1}(n-i) \\
&= c + cn(n-1) - c\sum_{i=1}^{n-1} i \\
&= c + cn(n-1) - c\frac{1}{2}(n-1)n \\
&= c + \frac{c}{2}n(n-1) \\
&= \frac{c}{2}n^2 - \frac{c}{2}n + c
\end{aligned}
$$

ここで

$$
\frac{T(n)}{n^2} = \frac{c}{2} - \frac{c}{2n} + \frac{c}{n^2} \le c
$$

であるため，計算量は $O(n^2)$ である．