アルゴリズム特論
順序統計量の計算

山本修身

最小値の計算

* 入力：配列 A (A[0], A[1], ..., A[n - 1])

n - 1回の比較で最小値が求まる

```
min = A[0];
for (i = 0; i < n; i++){
    if ( A[ i ] < min )
        min = A[ i ];
}
output( min );
```

最小値以外の「…番目」というような量を計算するにはどうすればよいか？

平均線形時間の順序統計量の計算 (1)

* 平均的に線形時間で良いのであれば，比較的簡単に求めることが可能である．

* 以下は配列のp番目からr番目までをある値より大きい部分と小さい部分に分けるプログラムである．

```
randomized_partition(A, p, r){
    i ← random(p, r);
    A[ p ] ⇄ A[ i ];
    x ← A[ p ]
    i ← p − 1; j ← r + 1;
    while ( 1 ){
        do { j ← j − 1;} while (A[ j ] > x)
        do { i ← i + 1;} while (A[ i ] <= x)
        if ( i < j )
            A[ i ] ⇄ A[ j ];
        else
            return j;
    }
}
```

平均線形時間の順序統計量の計算 (2)

* i番目の順序統計量を計算するプログラムは以下の通り：

```
randomized_select(A, p, r, i){
    if (p == r) return A[ p ];
    q ← randomized_partition(A, p, r);
    k = q - p + 1;
    if i <= k then
      return randomized_select(A, p, q, i);
    else
      return randomized_select(A, q + 1, r, i - k);
}
```

この場合最悪実行時間は $\Theta(n^2)$

平均実行時間の解析 (1)

* 平均実行時間の解析は以下のとおり.
* T(n)をn個の要素の選択の平均実行時間とする

$$T(n) \leq \frac{1}{n}(T(\max(1, n-1)) + \sum_{k=1}^{n-1} T(\max(k, n-k))) + O(n)$$

$$\leq \frac{1}{n}\left(T(n-1) + 2 \cdot \sum_{k=\lceil n/2 \rceil}^{n-1} T(k)\right) + O(n)$$

$O(n^2)/n = O(n)$

$$= \boxed{\frac{T(n-1)}{n}} + \frac{2}{n} \cdot \sum_{k=\lceil n/2 \rceil}^{n-1} T(k) + O(n)$$

$$= \frac{2}{n} \cdot \sum_{k=\lceil n/2 \rceil}^{n-1} T(k) + O(n)$$

平均実行時間の解析 (2)

* $k = 1, \ldots, n-1$ について $T(k) \leq ck$ と仮定する

$$T(n) = \frac{2}{n} \cdot \sum_{k=\lceil n/2 \rceil}^{n-1} ck + O(n)$$

$$= c\frac{2}{n}\left(\frac{n(n-1)}{2} - \frac{n/2(n/2+1)}{2}\right) + O(n)$$

$$= \frac{3}{4}cn + O(n) < cn + O(n)$$

数学的帰納法より，すべてのnについて$O(n)$であるといえる.