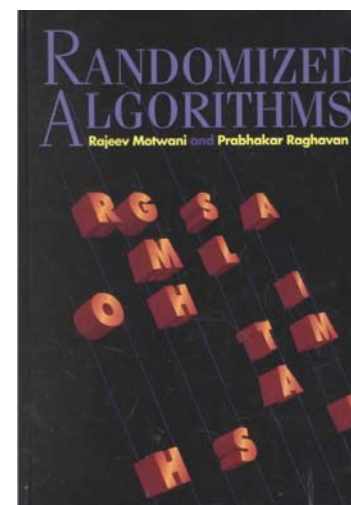


実践的幾何アルゴリズム

Advanced Algorithms for Computational Geometry

10.5. 乱択アルゴリズム:補足

担当: 上原隆平



Advanced Algorithms for Computational Geometry 実践的幾何アルゴリズム

10.5. Randomized Algorithms: Supplemental

Ryuhei Uehara

確率解析の例: QuickSortの解析

- ソーティング問題

Input: n 個のデータを記録した配列 $a[n]$

Output: 以下の条件を満たす配列 $a[n]$

$$a[1] < a[2] < \dots < a[n]$$

★話を単純にするため、 $a[i]=a[j]$, $i \neq j$ を満たすペアはないと仮定

- QuickSort は実用上、最速と言われることが多い
 - 典型的な分割統治法に基づくアルゴリズム
 - 都合良く分割されると $O(n \log n)$ 時間で計算が終わる
 - いつでも最悪の場合だと $O(n^2)$ かかる
 - ...理論的な解析と速度保証はできるのか？

確率解析の例: QuickSortの解析

- QuickSortのおさらい
 - $\text{qsort}(a, 1, n)$ を呼び出す
 - $\text{qsort}(a, i, j)$ が呼び出されると、
 - pivot $a[m]$ を(ランダムに)選ぶ
 - a を $a[m]$ を基準に「前半」と「後半」に分ける。つまり
 - $i \leq i' < m$ なら $a[i'] < a[m]$
 - $m < j' < j$ なら $a[j'] > a[m]$を満たすように並べ替える
 - $\text{qsort}(a, i, i'), a[m], \text{qsort}(a, j', j)$ がソート結果
- QuickSort は実用上、最速と言われることが多いが、、、
 - $a[m]$ がいつでも $a[i]..a[j]$ の中央の値だと
$$T(n) \leq 2T(n/2) + (c+1)n$$
が成立するので、 $T(n) = O(n \log n)$ を得る。
 - $a[m]$ がいつでも $a[i]$ や $a[j]$ だと
$$T(n) \leq T(1) + T(n-1) + (c+1)n$$
が成立するので、 $T(n) = O(n^2)$ を得る。

ラスベガスタイプ
のアルゴリズム

[余談]
毎回 $O(j-i)$ 時間かければ
ちょうど中央の値を
見つけることもできる。

平均的な場合
はどうなのか？

確率解析の例: QuickSortの解析

- QuickSort は実用上、最速と言われることが多いが、、、
 - 平均的には、 $a[i] \dots a[j]$ の値が一様に選ばれると仮定する。
 - つまり k 番目のものを pivot にする確率は $1/(j-i+1)$

[定理]

上記の仮定のもとでの QuickSort の実行時間の期待値の上界は
 $2n H(n) \sim 2n \log n$

- 記法

- $a[1] \dots a[n]$ の中で k 番目に来るべき要素を s_k と書く。
- 指示変数 (indicator variable) X_{ij} を以下のように定義する

$$X_{ij} = \begin{cases} 0 & s_i \text{ と } s_j \text{ がアルゴリズム中で比較されないとき} \\ 1 & s_i \text{ と } s_j \text{ がアルゴリズム中で比較されるとき} \end{cases}$$

- QuickSort の実行時間~要素の比較回数 = $\sum_{i=1}^n \sum_{j>i} X_{ij}$

オーバーヘッドの
少なさから、確かに
速いと言えそう

確率解析の例: QuickSortの解析

[定理]

QuickSort の実行時間の期待値の上界は $2n H(n) \sim 2n \log n$

- QuickSort の実行時間の期待値 = $E[\sum_{i=1}^n \sum_{j>i} X_{ij}] = \sum_{i=1}^n \sum_{j>i} E[X_{ij}]$ (期待値の線形性による)
- 「 p_{ij} : s_i と s_j が比較される確率」と定義すると、
$$E[X_{ij}] = p_{ij} \times 1 + (1 - p_{ij}) \times 0 = p_{ij}$$
よって p_{ij} の値を考える
- s_i と s_j はどんなときに比較されるのか？
 1. どちらかが pivot に選ばれている
 2. それまでの計算過程で、別々の qsort にわけられていない
 $\Leftrightarrow s_i$ と s_j の間の要素が、まだ pivot として選ばれていない

確率解析の例: QuickSortの解析

[定理]

QuickSort の実行時間の期待値の上界は $2n H(n) \sim 2n \log n$

• s_i と s_j はどんなときに比較されるのか？

1. どちらかが pivot に選ばれている
2. それまでの計算過程で、別々の qsort にわけられていない
 $\Leftrightarrow s_i$ と s_j の間の要素が、まだ pivot として選ばれていない

1. $s_i, s_{i+1}, s_{i+2}, \dots, s_{j-1}, s_j$ が pivot に選ばれる順番は、すべて等確率！

2. よってこれらの中で s_i か s_j が最初の pivot になる確率: $\frac{2}{j-i+1}$

つまり QuickSort の実行時間の期待値=

$$E\left[\sum_{i=1}^n \sum_{j>i} X_{ij}\right] = \sum_{i=1}^n \sum_{j>i} E[X_{ij}] = \sum_{i=1}^n \sum_{j>i} p_{ij} = \sum_{i=1}^n \sum_{j>i} \frac{2}{j-i+1} = \sum_{i=1}^n \sum_{k=2}^{n-i+1} \frac{2}{k} \leq 2 \sum_{i=1}^n \sum_{k=1}^n \frac{1}{k} = 2nH(n)$$