コンピュータサイエンス第二 課題 3「ソート」

概要

数あるソートアルゴリズムの特徴を調べるために、それぞれのアルゴリズムに異なる長さ、性質の数列を渡して、比較回数および実行時間を評価した。

第一部

課題 (a)

ソートする列の長さを適切に設定して各アルゴリズムと列種類について複数回実行して、長さと比較回数と 実行時間をエクセルファイル (sort-exam.xlsx) に記入した。以下では、得られたデータ全体から読み取れる各 アルゴリズムの特徴を考察する。

1 バブルソート

乱数列、正順列、逆順列どれを渡しても、列の長さが同じなら比較回数は変わらなかった。列の長さが 2 倍、3 倍…となると、比較回数は 2^2 倍、 3^2 倍…となることがわかった。

また、バブルソートでは比較回数と実行時間が比例することを実験的に確認した。 つまり、他のソートアルゴリズムと比べて、安定して比較回数が多く、遅いことがわかる。

2 挿入ソート

同じ列の長さなら、正順列を渡したとき比較回数は最小に、逆順列を渡したとき比較回数は最大になった。列の長さが 2 倍、3 倍…となると、正順列の場合の比較回数は 2 倍、3 倍…となるが、逆順列の場合は 2^2 倍、 3^2 倍…となった。乱数列の場合はその間となる。すなわち、渡す数列が正順列に近い場合は挿入ソートを使用すると良い。

また、長さ 100 の乱数列を渡したときの比較回数の変動係数はおよそ 0.06324 であった。上に述べた通り、 乱数列が正順列に近いかどうかで、比較回数が変わるため、変動が大きいことがわかる。

挿入ソートでは比較回数と実行時間が比例することを実験的に確認した。

3 マージソート

同じ列の長さなら、正順列、逆順列を渡したとき、比較回数は乱数列のものより小さくなった。列の長さが 2倍、3倍…となると、どの数列においても比較回数は2倍、3倍…より少し多くなる。

マージソートでは比較回数と実行時間は比例の関係ではないが、比較回数が増えるほど実行時間も増えることがわかった。

長さ 100 の乱数列を渡したときの比較回数の変動係数はおよそ 0.01086 であった。今回調べたアルゴリズムの中で一番小さい値となる。

4 クイックソート

同じ列の長さなら、正順列および逆順列を渡したとき比較回数は最大になった。このとき、列の長さが 2^{2} 倍、3 倍…となると、比較回数は 2^{2} 倍、 3^{2} 倍…となる。乱数列の場合はマージソートと同じような比較回数の増え方になった。

クイックソートでは比較回数と実行時間は比例の関係ではないが、比較回数が増えるほど実行時間も増える ことがわかった。

長さ 100 の乱数列を渡したときの比較回数の変動係数はおよそ 0.1105 であった。今回調べたアルゴリズムの中で一番大きい値となる。

第Ⅱ部

課題 (b)

長さ1億の数列をソートするとき、最も実行時間が長い、もしくは短いアルゴリズムを考える。また、その 実行時間を推測する。

5 乱数列

5.1 最長実行時間

バブルソート: $9.43 \times 10^8 s$

5.2 最短実行時間

クイックソート:?

6 正順列

6.1 最長実行時間

バブルソート: 9.43×10^8 s

6.2 最短実行時間

挿入ソート:20s

7 逆順列

7.1 最長実行時間

バブルソート: $9.43 \times 10^8 s$

7.2 最短実行時間

マージソート:?