データ構造とアルゴリズム  
演習課題2

提出日2024/05/25

HI4 45号 山口惺司

## レポート課題

# アルゴリズムの説明：

ダブルセレクションソートには、単純に最大値選択法と最小値選 択法を組み合わせた方法以外に、次のようなアルゴリズムもある。

①｛２ ８ ３ ４ ７ ６ １ ５｝が与えられたとする（ただし、偶数個の 場合のみ）。

②｛２ ８ ３ ４｝｛７ ６ １ ５｝の2つに分割する。

③｛２ ６ １ ４｝｛７ ８ ３ ５｝ … 2つのグループで先頭から1個ずつ取り出して大小を比較して、左側が大きい時は入れ替える（水色はグループ をまたいだ入れ替え）。

④１｛６ ２ ４｝｛７ ５ ３｝８ … 左のグループの最小値、右のグルー プの最大値を求め、それぞれ先頭、末尾と入れ替えて、グループから外す（緑 色はグループ内の入れ替え）。

⑤１｛６ ２ ３｝｛７ ５ ４｝８

⇒１ ２｛６ ３｝｛４ ５｝７ ８ … 2つのグループについて③、④を行う。

⑥１ ２ ｛４ ３｝｛６ ５｝７ ８

⇒１ ２ ３ ｛４｝｛５｝６ ７ ８

⇒１ ２ ３ ４ ５ ６ ７ ８ … 各グループとも1個になったら終了。

# 課題

1. この方法について、{8 4 3 9 6 5}がソートされる経過を図に示せ。また、比較と交換の回数も書け。

与えられた配列がソートされる経過を図1に示す。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 配列 | | | | | | 比較回数 | 交換回数 | 操作 |
| {8 | 4 | 3 | 9 | 6 | 5} |  |  | ① |
|  |  |  |  |  |  |  |  |
| {8 | 4 | 3} | {9 | 6 | 5} |  |  | ② |
|  |  |  |  |  |  |  |  |
| {8 | 4 | 3} | {9 | 6 | 5} | 3 | 0 | ③ |
|  |  |  |  |  |  |
| {8 | 4 | 3} | {9 | 6 | 5} | 4 | 2 | ④ |
|  |  |  |  |  |  |
| 3 | {4 | 8} | {5 | 6} | 9 | 2 | 1 | ③ |
|  |  |  |  |  |  |
| 3 | {4 | 6} | {5 | 8} | 9 | 2 | 0 | ④ |
|  |  |  |  |  |  |
| 3 | 4 | {6} | {5} | 8 | 9 | 1 | 1 | ③ |
|  |  |  |  |  |  |
| 3 | 4 | 5 | 6 | 8 | 9 |  |  |  |

図1 {8 4 3 9 6 5}がソートされる経過

図1より今回与えられた配列をソートするときの比較回数は12回、交換回数は4回となった。

1. この方法について、フローチャートと疑似コードを書き、Pythonで実装して結果を確認せよ。

フローチャート：

上記のアルゴリズムからフローチャートを作成し、図2に示す。

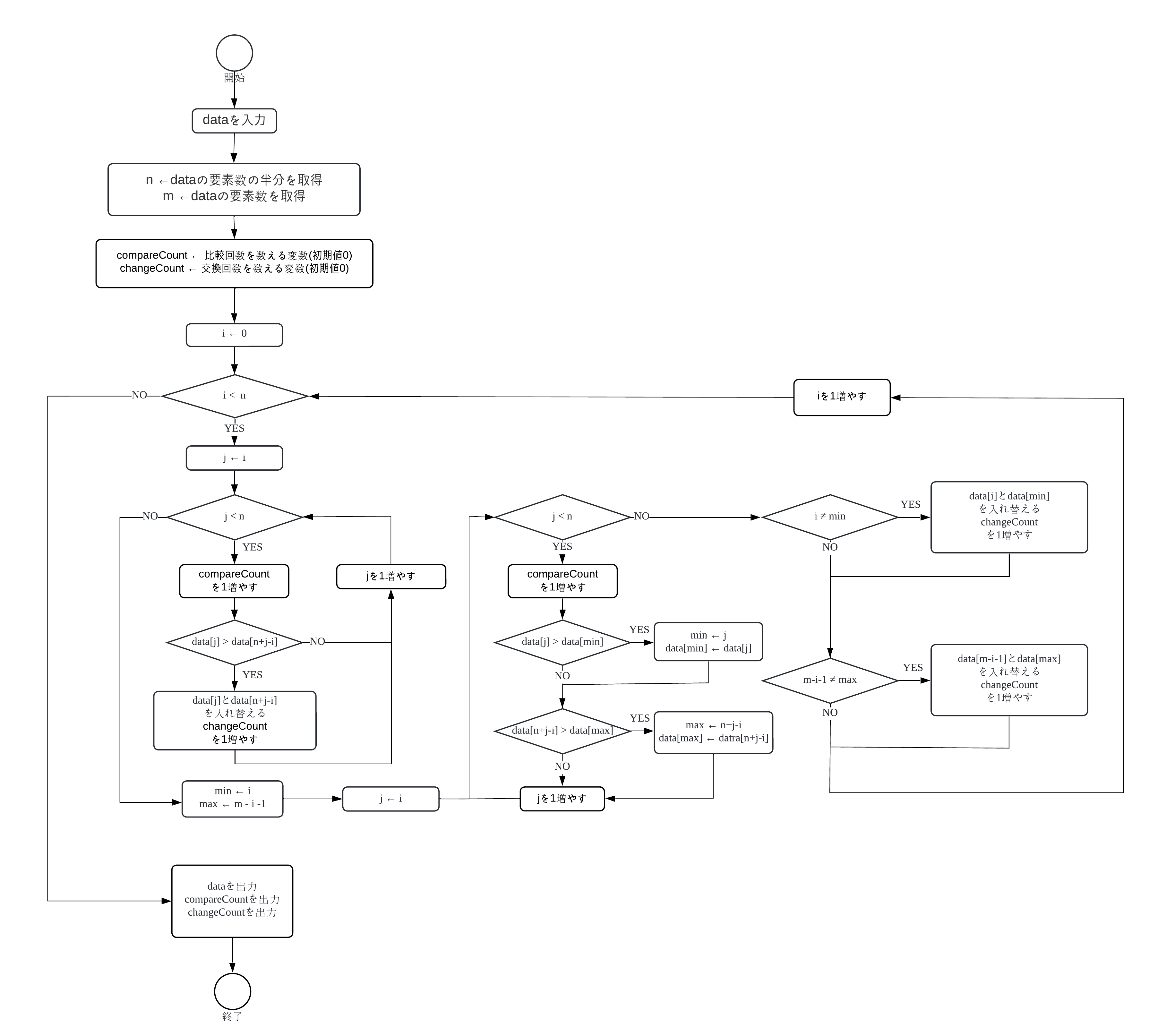


図2 ダブルセレクションソートフローチャート

疑似コード：

フローチャートから疑似コードを以下に示す。

1. ソートするデータdataを取得する。
2. 配列の半分の長さn, 配列の長さmを取得する
3. 比較回数と交換回数を数えるcompareCount, changeCount = 0を取得する。
4. i = 0を取得する。
5. i < nでなければ⑳に進む
6. j = iを取得する。
7. j<nでなければ⑪に進む
8. compareCountを1増やす
9. data[j] > data[n+j+i]であれば2つを入れ替え、changeCountを1増やす。そうでなければそのまま進む。
10. jを1増やし⑦に戻る
11. min=i, max=n, j=iを取得する。
12. j<nでなければ⑰に進む
13. compareCountを1増やす
14. data[j]>data[min]であればmin=j, data[min]=data[j]を取得する。そうでなければそのまま進む。
15. data[n+j-i]>data[max]であればmax=n+j-i, data[max]=data[j]を取得する。そうでなければそのまま進む。
16. jを1増やし⑫に戻る
17. i≠minであれば、data[i]とdata[min]を入れ替え、changeCountを1増やす。そうでなければそのまま進む
18. m-i-1≠maxであれば、data[m-i-1]とdata[max]を入れ替え、changeCountを1増やす。そうでなければそのまま進む
19. iを1増やし、⑤に戻る
20. data, compareCount, changeCountを出力して終了する

Pythonプログラム：

疑似コードを基にPythonプログラムを作成し、以下に示す。

data = [8, 4, 3, 9, 6, 5]

n = int(len(data)/2)

m = int(len(data))

print("ソート前：", data)

changeCount = 0

compareCount = 0

for i in range(n):

    for j in range(i, n):

        compareCount += 1

        if data[j] > data[n + j - i]:

            data[j], data[n + j - i] = data[n + j -i], data[j]

            changeCount += 1

    min = i

    max = m - i - 1

    for j in range(i, n):

        compareCount += 1

        if data[j] < data[min]:

            min = j

            data[min] = data[j]

        if data[n + j - i] > data[max]:

            max = n + j -i

            data[max] = data[n + j - i]

    if i != min:

        data[i], data[min] = data[min], data[i]

        changeCount += 1

    if (m-i-1) != max:

        data[m-i-1], data[max] = data[max], data[m-i-1]

        changeCount += 1

print("ソート後：", data)

print("比較回数：", compareCount)

print("交換回数：", changeCount)

実行結果：

　上のプログラムを実行した時の実行結果を図3に示す。

時計 が含まれている画像

自動的に生成された説明

図3 {8 4 3 9 6 5}が与えられた時の実行結果

　また、与える配列を変えて実行した時の実行結果を図4～8に示す。

テキスト が含まれている画像

自動的に生成された説明

図4 {2 10 1 4 5 5 8 6}が与えられた時の実行結果

文字が書かれている

中程度の精度で自動的に生成された説明

図5 {1 2 3 4 5 6 7 8 9 10}が与えられた時の実行結果

時計 が含まれている画像

自動的に生成された説明

図6 {6 -1 -4 3 11 2 0 28}が与えられた時の実行結果

時計 が含まれている画像

自動的に生成された説明

図7 {5 2 3 1 4}が与えられた時の実行結果

カレンダー が含まれている画像

自動的に生成された説明

図8 {2 2 2 2 2 2}が与えられた時の実行結果

# 考察

図7は配列の要素数を奇数にした場合だが、要素数が奇数個だとうまく動作しないことが分かった。

図3～図6は全て正しくソートできているため、プログラムは正しく実装できていると言える。

図3と図8から、ダブルセレクションソートは、同じ要素数の配列でも与えられた配列によって交換回数が違うということがわかった。

また、比較回数は表1のようになる

表1 要素数と比較回数の関係

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 要素数 | 0 | 2 | 4 | 6 | 8 | 10 |
| 比較回数 | 0 | 2 | 6 | 12 | 20 | 30 |

つまり要素数nにおける比較回数は以下の式で表せる。