プログラミング演習I 第9回 演習レポート

2164204 野村英吉

(1) 基本課題2

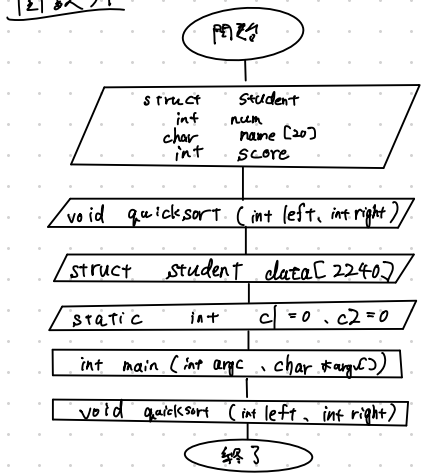
2164204_b2.c

整列過程におけるデータの比較回数と

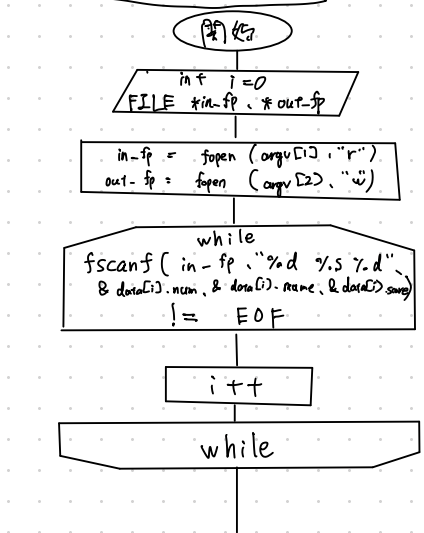
データの入れ換え回数を表す。

(2)

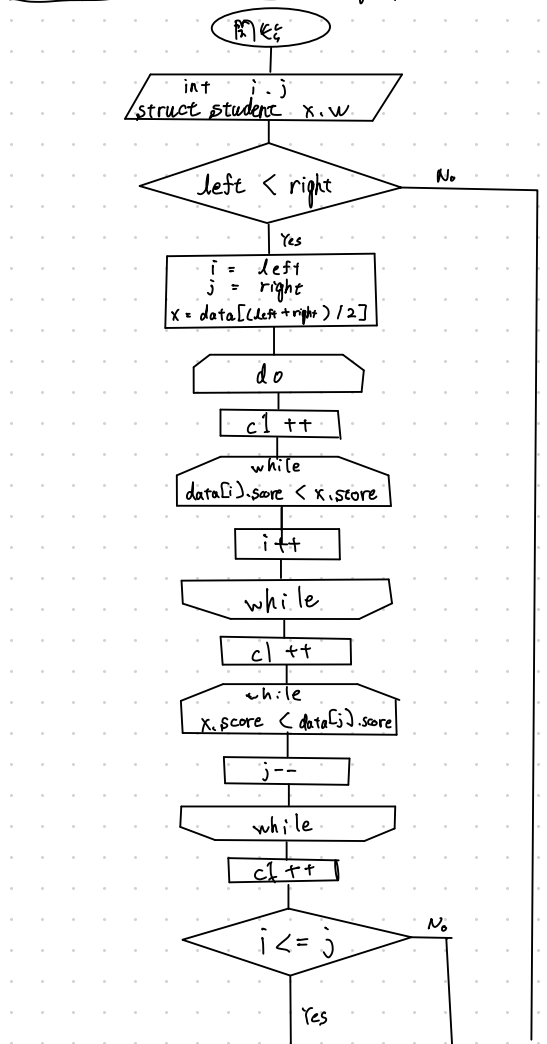
関数外

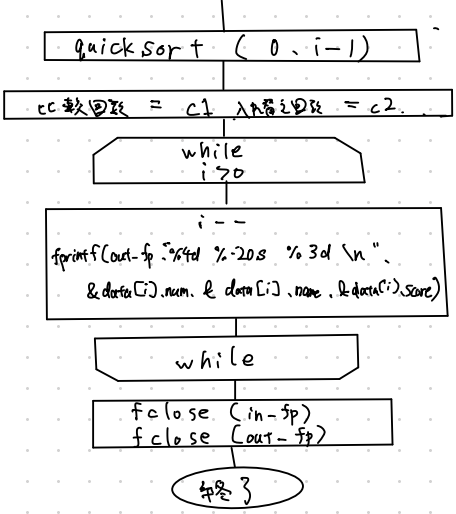


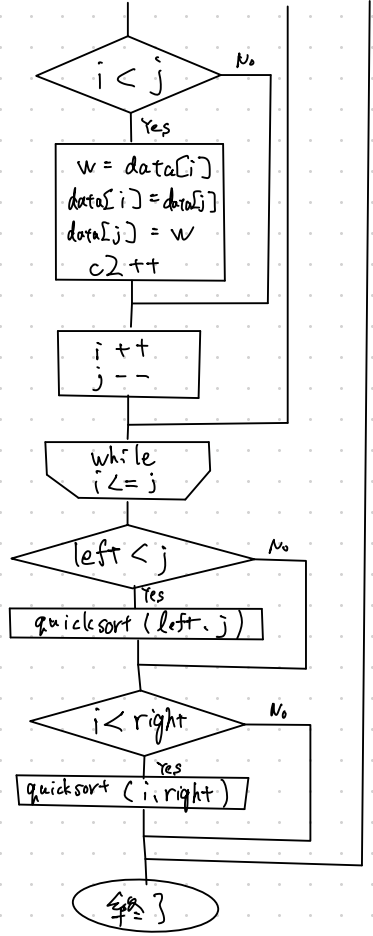
int main (int argc, char *argv[])



void quicksort (int left, int right)







(3) quicksort(int left, int right)について考える.

基本的には left < right が成り立つとする.

i = left, j = right, x = data[(left + right)/2] とする.

data[i].score と x.score をcc数.

data[i].score < x.score となるまで i の値を増やす.

これが1回のcc数.

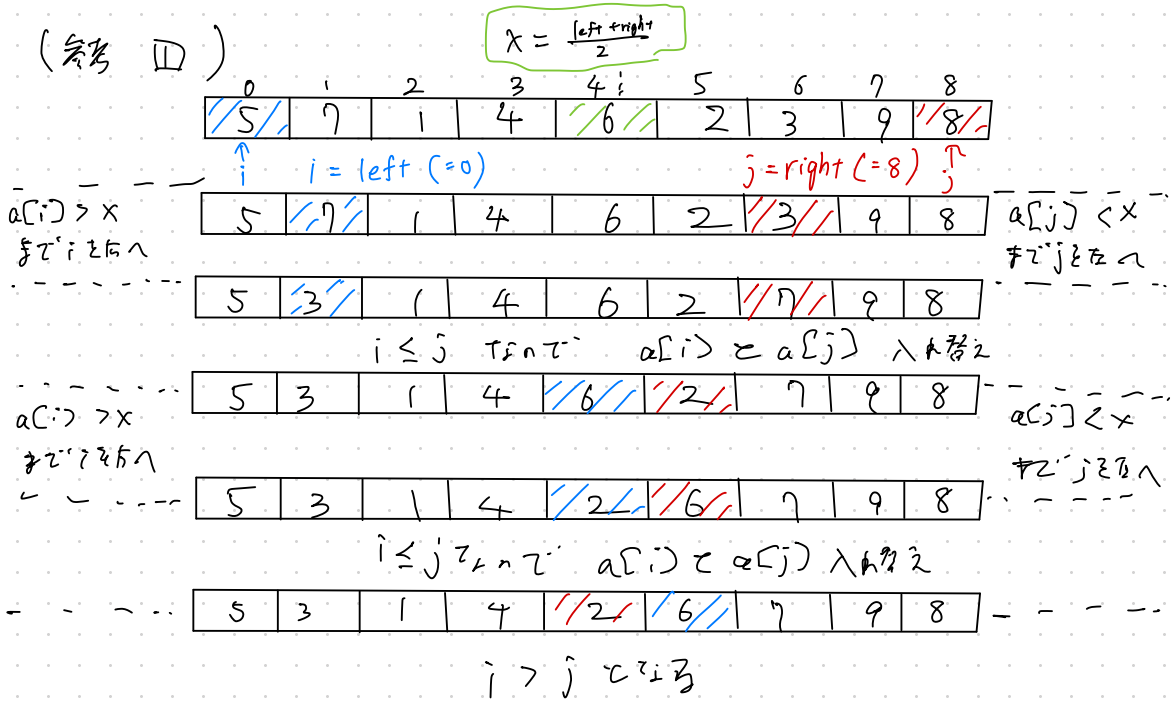
次に, data[j].score と x.score をcc数.

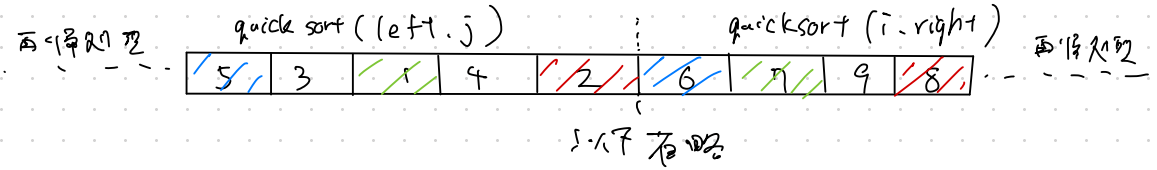
x.score < data[j].score となるまで j の値を減らす.

これが1回のcc数.

この処理が終わった後、i<jなら
data[i]とdata[j]を入れ替え。これが1回の入れ替え。
その後 iを1増やし、jを1減らす。
上記の処理を i≦jの関係が崩れないまでやる。
i>jとなった後を考える。
left<j (jが減りすぎてleftより下になってない場合)
なら、quicksort(left, j)で再帰処理。
i<right (iが増えすぎてrightより上になってない場合)
なら、quicksort(i, right)で再帰処理。
次第にi, j, left, rightがi≒j/2付近に収束し、
比較、入れ替えが終わる。

(参考 図)





クイックソートでは、両こうで標準体の中央値 x を基準に、
a[i] > x となるまで i カーソルを右に、a[j] < x となるまで
j カーソルを左へ移動。その後 a[i] と a[j] を入れ替え、
さらに i カーソルを右に、j カーソルを左に、両方止まったら
値を入れ替え。 i > j となったとき、左はしから j
までで、i からたはしまでで両こうに、標準体を分割。
以上の処理をくり返すと、最終的に分割された
両こうが分割しきれなくなり、そこまでですべての
入れ替えが完了する。

(4) 第8回で説明する内容は省略する。
関数外部に static int 型変数 c1=0、c2=0 を
定義。c1 は比較回数、c2 を入れ替え回数とする。
quicksort 関数について、i = left、j = right、
x = data[(left + right)/2] とする。
left < right の場合、以下の処理へ。
do while 文では、i ≦ j の間繰り返す。
1回の比較は、data[i].score と x.score で
data[i].score < x.score まで i を増やす。(c1++)

(回のに数は、data[j].scoreとx.scoreで
x.score < data[j].scoreまでjを減やす。(c1++)
iとjを比較。(c1++) i≦j かつ
i<j なら 入れ替え用のstudent型構造体変数w
を用いて、data[i]とdata[j]を入れ替える。(c2++)
i≦jなら i++、j--.
ここまでがdo whileえ。
do whileえを抜けると、j<iとなっている。
left<jなら quicksort(left, j)で再び呼ばれる。
i<right なら quicksort(i, right)で再び呼ばれる。

5) 今回は、クイックソートのアルゴリズムを用いた
データの入れ替えだった。前回よりも複雑で、
理解、実装に時間がかかった。実際、まだ
理解度が低いとレポートを書いて感じたが、
効率的なソート方法であることは体感できた。
マージソートについても、今回考察に時間を
さけなかったので、時間のあるときにじっくり
考察したい。

(6) 四：「明解C言語によるアルゴリズムとデータ構造」

P175、P176

2008、9、10　　奥田望洋、辻亮介

(7) 参考資料を用いることは自分の理解を高めるためにとても重用だと感じた。