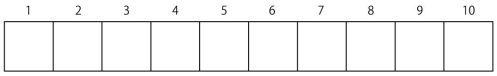
第8章　クイックソートの解説

このファイルでは、スタックを使ったクイックソートについて、図を交えて処理の様子を説明します。スタックのはたらきを理解することが目的ですので、比較、交換などの処理については省略しています。

（動作の確認用そして「sample8_スタック確認.sb2」というファイルを用意しています。詳しくはこの文書の最後のページの説明をみてください。）

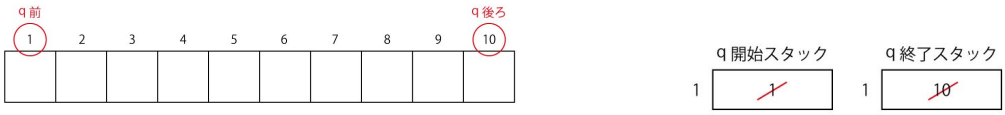
①　最初の状態です。要素数が10のリスト（配列）で説明します。要素は省略しています。



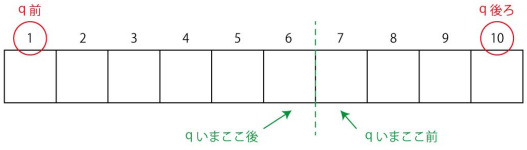
②　「q開始スタック」と「q終了スタック」にそれぞれ1と最後の添字（この場合は10）が入ります。

③　「q開始スタック」のリストの長さは1です（つまり空ではない）ので、繰り返しを続けます。

④　「q前」「q後ろ」に、それぞれ「q開始スタック」「q終了スタック」の最後の要素を入れて、「q開始スタック」「q終了スタック」の最後の要素を削除します。

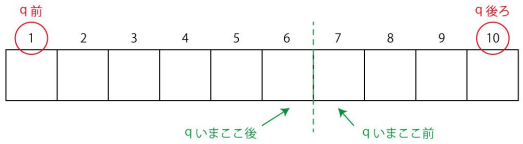


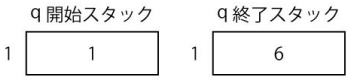
⑤　そのまま処理を続けていくと、いずれ次のように「qいまここ前」と「qいまここ後」が交差します。



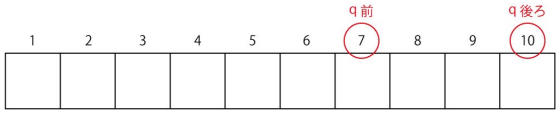
⑥　「qいまここ前-q前」と「q後ろ-qいまここ後」を比べます。これは、交差した箇所（下の図の緑の点線）を堺に、前半と後半の要素の数を比べるという意味です。前半は1～6で要素の数が6つ、後半は7～10で要素の数が４つ。したがって前半のほうが要素の数が多いので、これを（この場合は1～6）を「q開始スタック」と「q終了スタック」に記録しておきます

条件分岐のうち、こちらが実行される。（つぎの図のとおり）。

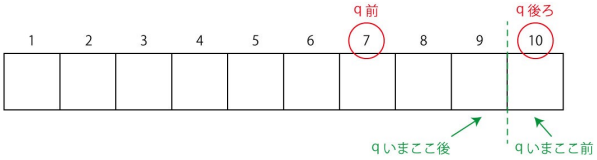




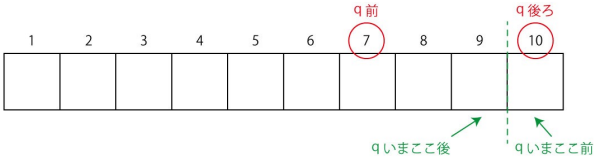
このように1～6をいったん記録しておいて、もう片方の要素の数が少ない方（この場合は7～10）の処理をすすめます。（上の図のブロックの「q前をqいまここ後+1にする」が実行されます）

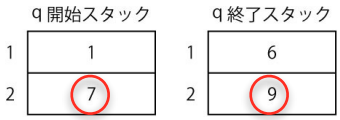


⑦　要素7を「q前」、要素10を「q後ろ」として、要素７～10の中で同じように処理がすすんでいきます。いずれ、図のように「qいまここ前」と「qいまここ後」が交差します。

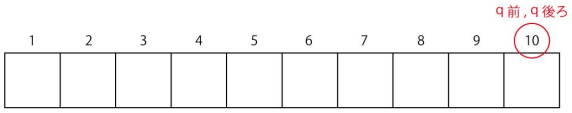


⑧　交差をしたら、先ほどと同じく、要素の数が多い方（この場合は7～9のほうが10～10よりも多い）をスタックに記録します。





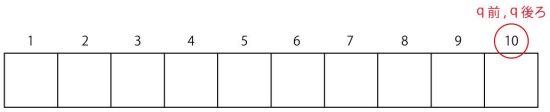
さきほどと同じく、要素の数が少ない方（この場合は10～10）に対して、新たに「q前」「q後ろ」が設定されます。



	q開始スタック
1	1
2	7

	q終了スタック
1	6
2	9

⑨　この時点で、q前 = q後ろとなり、「q前<q後ろではないまで繰り返す」という条件がfalseになるので、繰り返しが終わります。



繰り返しが終わるので、この中は実行されない

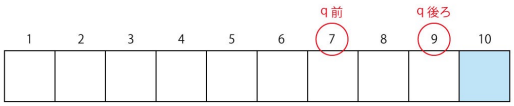
⑩　「q開始スタックの長さ=0」がチェックされます。

	q開始スタック
1	1
2	7

	q終了スタック
1	6
2	9

現在、「q開始スタック」の長さは0ではないので（要素が２つある。1と７）、繰り返しの中が実行されます。

⑪　「q前」「q後ろ」に、それぞれ「q開始スタック」「q終了スタック」の最後の要素を入れて、「q開始スタック」「q終了スタック」の最後の要素を削除します。



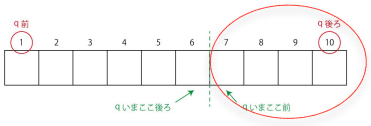
	q開始スタック
1	1
2	~~7~~

	q終了スタック
1	6
2	~~9~~

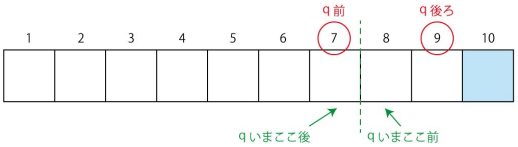
上の図で要素10は先ほど処理が終了したので確定しています（色を変えています）。

今度は、要素7～9で処理を行います。

この時点で、最初に２つに分けたうちの後ろ側のかたまりの処理がまだ続いていることに注意してください。



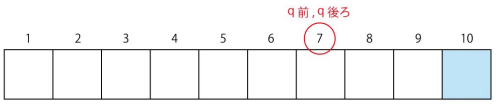
⑫　要素7～9で同様に処理を続けていくと、いずれまた、図のように「qいまここ前」と「qいまここ後」が交差します。



	q開始スタック
1	1

	q終了スタック
1	6

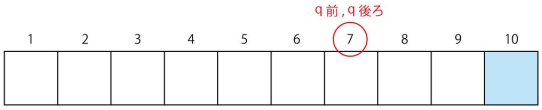
これまで同様に、要素の数が多い方（この場合は8～9）をスタックに記録し、そして、要素の数が少ない方（この場合は7～7）に対して、新たに「q前」「q後ろ」が設定されます。



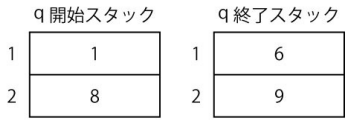
	q開始スタック
1	1
2	8

	q終了スタック
1	6
2	9

⑬　この時点で、「q前<q後ろではないまで繰り返す」という条件がfalseになるので、繰り返しが終わります。

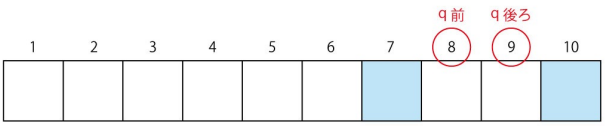


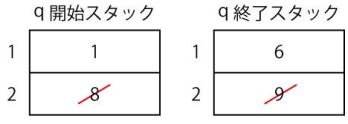
⑭　「q前<q後ろではないまで繰り返す」の繰り返しが終わったので、「q開始スタックの長さ=0」がチェックされます。



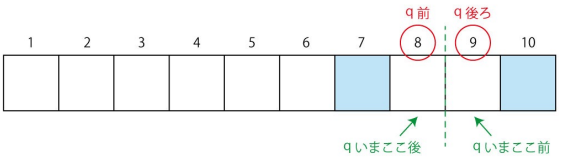
現在、「q開始スタック」の長さは0ではないので、繰り返しの中が実行されます。

⑮　「q前」「q後ろ」に、それぞれ「q開始スタック」「q終了スタック」の最後の要素を入れて、「q開始スタック」「q終了スタック」の最後の要素を削除します。

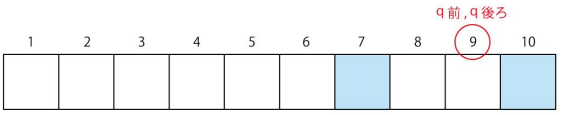




⑯　要素8～9で同様に処理を続けていくと、図のように「qいまここ前」と「qいまここ後」が交差します。



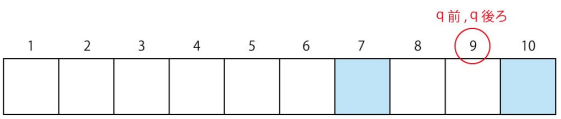
これまで同様に、要素の数が多い方と少ない方に分けたいのですが、両方とも同じ要素数です（1つずつ）。ブロックの条件（下図）に従うと、この場合は「qいまここ前-q前」と「q後ろ-qいまここ後」が同じなので、「でなければ」のほうが実行されます。つまり、8をスタックに記録し、新たに9が「q前」「q後ろ」が設定されます。



	q開始スタック
1	1
2	8

	q終了スタック
1	6
2	8

⑰　この時点で、「q前<q後ろではないまで繰り返す」という条件がfalseになるので、繰り返しが終わります。



	q開始スタック
1	1
2	8

	q終了スタック
1	6
2	8

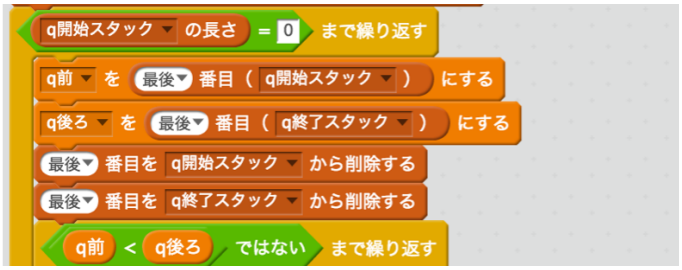
⑱「q前<q後ろではないまで繰り返す」の繰り返しが終わったので、「q開始スタックの長さ=0」がチェックされます。

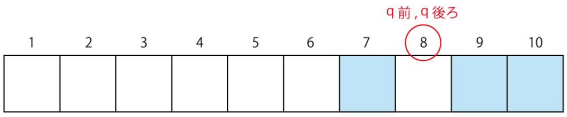
	q開始スタック
1	1
2	8

	q終了スタック
1	6
2	8

現在、「q開始スタック」の長さは0ではないので、繰り返しの中が実行されます。

⑲　「q前」「q後ろ」に、それぞれ「q開始スタック」「q終了スタック」の最後の要素を入れて、「q開始スタック」「q終了スタック」の最後の要素を削除します。

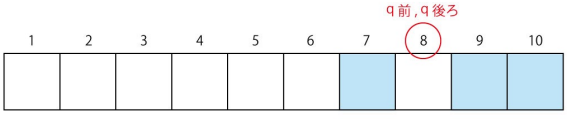




	q開始スタック
1	1
2	~~8~~

	q終了スタック
1	6
2	~~8~~

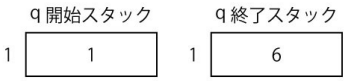
⑳　この時点で、「q前<q後ろではないまで繰り返す」という条件がfalseになるので、繰り返しが終わります。



	q開始スタック
1	1

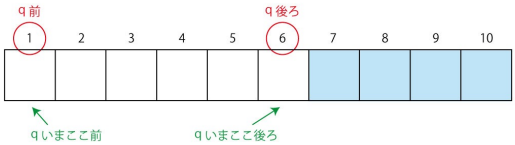
	q終了スタック
1	6

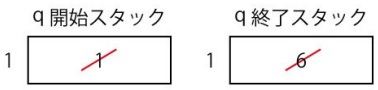
㉑「q前＜q後ろではないまで繰り返す」の繰り返しが終わったので、「q開始スタックの長さ=0」がチェックされます。



現在、「q開始スタック」の長さは0ではないので、繰り返しの中が実行されます。

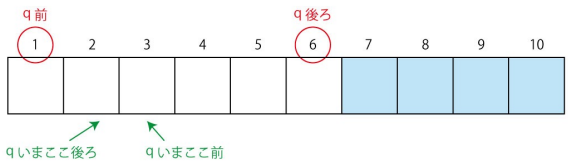
㉒　「q前」「q後ろ」に、それぞれ「q開始スタック」「q終了スタック」の最後の要素を入れて、「q開始スタック」「q終了スタック」の最後の要素を削除します。



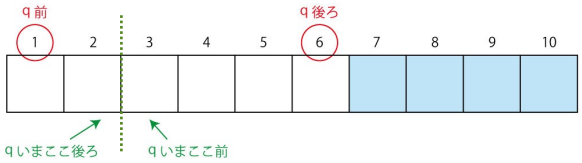


処理が、最初に２つに分けたうちの前半に移りました。

㉓　処理を続けていくと、図のように「qいまここ前」と「qいまここ後」が交差します。

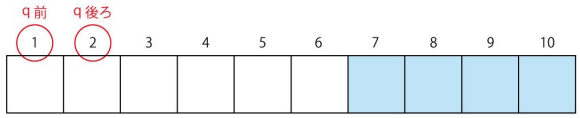


㉔　「qいまここ前-q前」と「q後ろ-qいまここ後」を比較します。後半（3～6）の要素数が多いので、これをスタックに記録します。

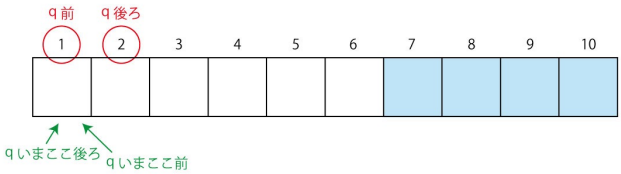


	q開始スタック		q終了スタック
1	3	1	6

そして、要素の数が少ない方（この場合は1～2）に対して、新たに「q前」「q後ろ」が設定されます。



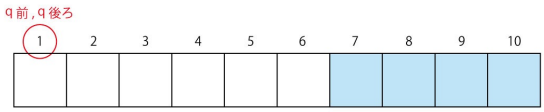
㉕　処理を続けていくと、「qいまここ前」と「qいまここ後」が交差します。データの並び方によっては、図のように「qいまここ後ろ」と「qいまここ前」が同じ要素を差して終わることもあります。スクリプトにしたがって、「q開始スタック」「q終了スタック」に後半を記録します（2～2）。



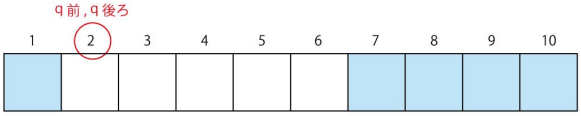
こちらが実行される

	q開始スタック		q終了スタック
1	3	1	6
2	2	2	2

さらに、要素の数が少ない方（この場合は1～2）に対して、新たに「q前」「q後ろ」が設定されます。



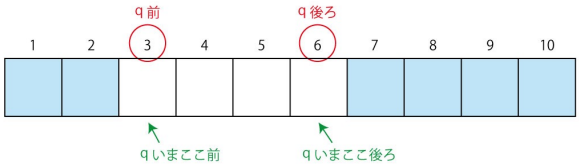
㉖　これまで同様に、要素が1つになったら（q前とq後ろが重なったら）そこで処理は終わりです。「q開始スタック」の長さが0でなければ（つまりまだスタックにデータがあれば）、「q開始スタック」と「q終了スタック」の最後からそれぞれデータを取り出して、あらたに「q前」「q後ろ」にそれぞれ入れます。



	q開始スタック
1	3
2	~~2~~

	q終了スタック
1	6
2	~~2~~

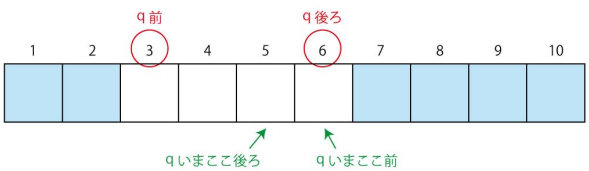
㉗　そのまま処理が行われますが、いきなり要素が1つ（2～2）なので、そこで処理が終わり、2番目の要素は確定です。今度は、「q開始スタック」「q終了スタック」から3と6がそれぞれ取り出され、新たに「q前」「q後ろ」に入ります。つまり、要素3～6の処理が始まったということです。



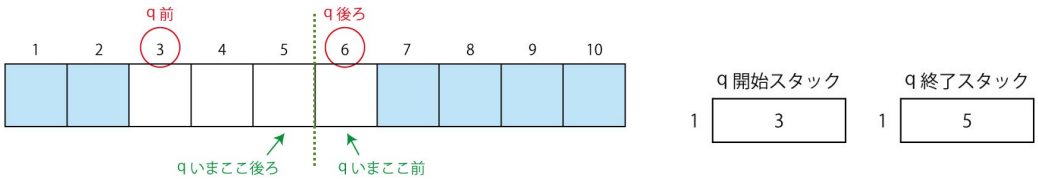
	q開始スタック
1	~~3~~

	q終了スタック
1	~~6~~

㉘　処理が進み、「qいまここ前」と「qいまここ後ろ」が交差しました。

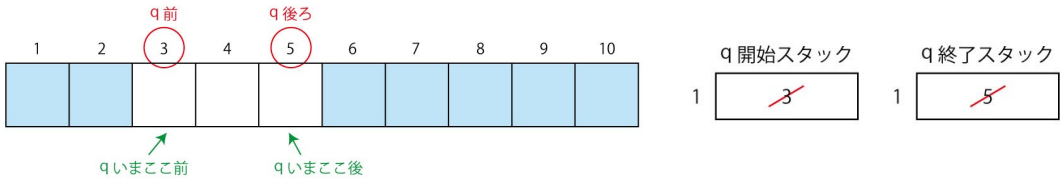


㉙　交差した箇所の前と後ろが比較され、要素の数が多い方（3〜5）がスタックに記録されます。



㉚　6〜6は要素が1つしかなく、「q前」と「q後ろ」が重なったので処理が終わります。

㉛　スタックからデータが取り出され、新たに「q前」「q後ろ」がセットされ、スタックの中身が削除されます。



長くなってきたのでこのあたりで説明を終わりますが、あとは、これまでと同様の仕組みで3〜5が処理されていきます。

＜まとめ＞

- スタック（「q開始スタック」「q終了スタック」）のはたらきは、ある固まりの処理をしている間に、他の固まりが、どの要素〜どの要素までだったのかを記録しておくためのものです。

- 今回の説明では要素が10個だけだったので、スタックは最大でも2つまでしかデータを記録しませんでしたが、要素数が多くなるとスタックに記録されるデータが増えていきます。

- 動作の確認用に、「sample8_スタック確認.sb2」というファイルを用意しました。これは、要素の数を100にしてあり、スタックにデータが追加されたときや、スタックからデータを取り出したときに、ネコがそれを報告します。ターボモードではなく通常のモードでクイックソートを実行すると、スタックの動作がじっくりと観察できます。