# Optimized Bubble Sort (Java)

10

4

I would like to know how else I can optimize bubble sort so that it overlooks elements that have already been sorted, even after the first pass.

```
Eg. [4, 2, 3, 1, 5, 6] --> [2, 3, 1, **4, 5, 6**]
```

We observe that [4,5,6] are already in sorted order, how can modify my code so that it overlooks this 3 elements in the next pass? (which means the sort would be more efficient?) Do you suggest a recursive method?

```
public static void bubblesort(int[] a) {
  for(int i=1; i<a.length; i++) {
    boolean is_sorted = true;

    for(int j=0; j<a.length; j++) {
      if(a[j] > a[j+1]) {
         int temp = a[j];
         a[j] = a[j+1];
         a[j+1] = temp;
         is_sorted = false;
      }
    }

    if(is_sorted) return;
  }
}
```

Thanks for your time!

java optimization recursion bubble-sort

share improve this question

asked Apr 24 '13 at 14:48

kent

93 1 1 9

add a comment

## 8 Answers

active oldest votes

16

First of all, you have an out-of-bounds access:

```
    for(int j=0; j<a.length; j++) {
      if(a[j] > a[j+1]) {
```

for `j == a.length-1`, so the loop condition should rather be `j < a.length-1`.

But, in Bubble sort, you know that after `k` passes, the largest `k` elements are sorted at the `k` last entries of the array, so the conventional Bubble sort uses

```
public static void bubblesort(int[] a) {
  for(int i=1; i<a.length; i++) {
    boolean is_sorted = true;

    for(int j=0; j < a.length - i; j++) { // skip the already sorted largest elements
      if(a[j] > a[j+1]) {
         int temp = a[j];
         a[j] = a[j+1];
         a[j+1] = temp;
         is_sorted = false;
      }
    }

    if(is_sorted) return;
  }
}
```

Now, that would still do a lot of unnecessary iterations when the array has a long sorted tail of largest elements, say you have `k,k-1,...,1` as the first `k` elements and `k+1` to `100000000` in order after that. The standard Bubble sort will pass `k` times through (almost) the entire array.

But if you remember where you made your last swap, you know that after that index, there are the largest elements in order, so

```
public static void bubblesort(int[] a) {
  int lastSwap = a.length-1;
  for(int i=1; i<a.length; i++) {
    boolean is_sorted = true;
    int currentSwap = -1;

    for(int j=0; j < lastSwap; j++) {
      if(a[j] > a[j+1]) {
         int temp = a[j];
         a[j] = a[j+1];
         a[j+1] = temp;
         is_sorted = false;
         currentSwap = j;
      }
    }

    if(is_sorted) return;
    lastSwap = currentSwap;
  }
}
```

would sort the above example with only one pass through the entire array, and the remaining passes only through a (short) prefix.

Of course, in general, that won't buy you much, but then optimising a Bubble sort is a rather futile exercise anyway.

share improve this answer

answered Apr 24 '13 at 15:31

Daniel Fischer
**164k** 14 268 399

1

appreciate your detailed explanation as well as spotting that jarring error of mine! – kent Apr 24 '13 at 15:42

it is a bit cleaner/more clear to use a while loop for the outer loop and check the `currentSwap` variable. – employee-0 Sep 14 '13 at 17:57

I did not know the last optimization for the long ordered tail, thanks. – Mindaugas Bernatavičius Aug 22 '18 at 12:51

add a comment

1

```
 public static Integer[] optimizedbubbleSort(Integer[] input){
    long startTime = System.nanoTime();
    boolean swapped = true;
    for(int pass=input.length-1; pass>=0 && swapped; pass--){
        swapped = false;
        for(int i=0; i<pass; i++){
            if(input[i]>input[i+1]){
                int temp = input[i];
                input[i] = input[i+1];
                input[i+1] = temp;
                swapped = true;
            }
        }
    }
    System.out.println("Time taken for OPTIMIZED bubbleSort: "+(System.nanoTime() - startTime));
    return input;
}
```

share improve this answer

answered Jun 11 '15 at 4:04

Sharath
**140** 2 12

This is not optimized. You are only going in reverse and showing the time taken for the operation. – kbluue Oct 27 '17 at 10:02

add a comment

0

you should use a variable "size" for the inner loop and change it to the latest swapped element in each cycle.This way your inner loop goes up to the latest "swapped" element and passes the rest that are unswapped (aka in their correctplace). i.e

```
do {
        int newsize =0;
        for (int i = 1; i < size; i++) {
            if (a[i - 1] > a[i]) {
                int temp;
                temp = a[i - 1];
                a[i - 1] = a[i];
                a[i] = temp;
                newsize =i;
            }
        }
        size = newsize;
   } while (size > 0);
```

share improve this answer

answered Oct 19 '14 at 14:19

Panos Gr
88 • 7

add a comment

0

```
    public static void BubbleSorter(params int[] input){
        int newSize = input.Length-1, size = 0;
        bool swap;
        do
        {
            swap = false;
            for (int j = 0; j < newSize; j++)
            {
                if (input[j] > input[j + 1])
                {
                    int temp = input[j + 1];
                    input[j + 1] = input[j];
                    input[j] = temp;
                    swap = true;
                    size = j;
                }
            } newSize = size;
        } while (swap);

        DisplayArrayElements(input);
    }
```

share improve this answer

answered Nov 1 '15 at 5:10

Dakshitha Mevan Dias
3 • 2

This is a c# code Ive written for bubble sort – Dakshitha Mevan Dias Nov 1 '15 at 5:12

add a comment

0

I devised a method that reduces the number of iterations by excluding parts at the beginning and end of the array that have been ordered in previous loops.

```
static int[] BubbleSortOptimized(int arr[]) {
    int start = 0, stop = arr.length - 1, control = 0;
    boolean ordered, nsCaught;
    while (true){
        ordered = true;
        nsCaught = false;
        for (int i = start; i < stop; i++) {
            if (i > 1) {
                if (!nsCaught && arr[i-2] > arr[i-1]){
                    ordered = false;
                    start = i-2;
                    nsCaught = true;
                }
            }
            if (arr[i] > arr[i+1]){
                int hold = arr[i];
                arr[i] = arr[i+1];
                arr[i+1] = hold;
                control = i;
            }
        }
        System.out.println(Arrays.toString(arr));
        if (ordered) return arr;
        stop = control;
    }
}
```

But as @Daniel Fischer said in an earlier answer, it doesn't do a lot with larger arrays.

share improve this answer

answered Oct 27 '17 at 10:40

kbluue
98 • 13

add a comment

0

In the above example, the array got sorted after 3rd pass, but we will still continue with the 4th, 5th pass. Suppose if the array is already sorted, then there will be no swapping (because adjacent elements are always in order), but still we will continue with the passes and there will still be (n-1) passes.

If we can identify, that the array is sorted, then we should stop execution of further passes. This is the optimization over the original bubble sort algorithm.

If there is no swapping in a particular pass, it means the array has become sorted, so we should not perform the further passes. For this we can have a flag variable which is set to true before each pass and is made false when a swapping is performed.

```
void bubbleSort(int *arr, int n){
for(int i=0; i<n; i++)
{
  bool flag = false;
   for(int j=0; j<n-i-1; j++)
   {
      if(array[j]>array[j+1])
      {
        flag = true;
         int temp = array[j+1];
         array[j+1] = array[j];
         array[j] = temp;
      }
   }
  // No Swapping happened, array is sorted
  if(!flag){
     return;
  }}}
```

share improve this answer

answered May 20 '18 at 18:35

Chamila Maddumage
878 • 1 • 9 • 23

add a comment

0

```
public class Tester {
    static boolean bubbleFlag = true;

    public static void main(String[] args) {
        int array[] = new int[] {
            1,
            9,
            2,
            3,
            4,
            5,
            6
        };
        bubbleSort(array);
    }

    private static void bubbleSort(int...array) {
        System.out.println("Before Sorting: " + Arrays.toString(array));

        for (int i = 0; i < array.length - 1; i++) {
            if (i > 0) if (bubbleFlag) break;

            for (int j = 0; j < array.length - i - 1; j++) {
                if (array[j] > array[j + 1]) array = swap(j, j + 1, array);
                System.out.println("Iteration " + i + " :" + Arrays.toString(array));
            }
            bubbleFlag = true;
        }
    }

    private static int[] swap(int i1, int i2, int...is) {
        bubbleFlag = false;
        is[i1] = is[i1] + is[i2];
        is[i2] = is[i1] - is[i2];
        is[i1] = is[i1] - is[i2];
        return is;
    }
```

share improve this answer

edited Jul 26 '18 at 7:41

Tim Diekmann
3,698 • 9 • 20 • 40

answered Jul 26 '18 at 7:24

Ahmed meeran
43 • 5

add a comment

-1

**Optimized bubble sort with just 1 for Loop**

```
/*Advanced BUBBLE SORT with ONE PASS*/
/*Authored by :: Brooks Tare  AAU*/

public class Bubble {

    public int[] bubble(int b[]){
    int temp,temp1;

    for(int i=0;i<b.length-1;i++){

            if(b[i]>b[i+1] ){
                ///swap(b[i],b[i+1]);

                temp=b[i];
                b[i]=b[i+1];
                b[i+1]=temp;

    /*Checking if there is any number(s) greater than
      the current number. If there is swap them.*/
                while(i>0){


                    if(b[i]<b[i-1]){
                    ///swap(b[i]<b[i-1])

                        temp1=b[i];
                        b[i]=b[i-1];
                        b[i-1]=temp1;
                        i--;
                    }
                    else if(b[i]>b[i-1]){i--;}
                }
            }
            else{continue;}

        }
```

share improve this answer

answered Aug 17 '18 at 15:01

Brook tare
1

add a comment

## Your Answer

B

## Sign up or log in

Sign up using Google | Sign up using Facebook | Sign up using Email and Password

## Post as a guest

**Name**

**Email**
Required, but never shown

Post Your Answer

*By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy*

Not the answer you're looking for? Browse other questions tagged java optimization recursion bubble-sort or ask your own question.

asked 6 years, 1 month ago

viewed 23,323 times

active 9 months ago

**Blog**

- Stack Overflow and Pursuit: Nurturing A New Generation of Developers

**Featured on Meta**

- Unicorn Meta Zoo #4: What makes for a healthy community?

## Linked

2
Bubble sort array, how can I make this php bubble sort code better or more effective?

0
is there a more sophisticated version of bubble sort algorithm in java?

-1
Optimizing bubble sort - What am I missing?

0
Bubble sort implementation from Pseudocode

0
Optimizing BubbleSort

## Related

5978
Is Java “pass-by-reference” or “pass-by-value”?

2945
How do I efficiently iterate over each entry in a Java Map?

2174
Does a finally block always get executed in Java?

2848
What is the difference between public, protected, package-private and private in Java?

3677
How do I read / convert an InputStream into a String in Java?

2842
When to use LinkedList over ArrayList in Java?

3225
How do I generate random integers within a specific range in Java?

2786
How do I convert a String to an int in Java?

2958
Creating a memory leak with Java

23103
Why is processing a sorted array faster than processing an unsorted array?

## Hot Network Questions

- Why can't we feel the Earth's revolution?
- Why are backslashes included in this shell script?
- Is there a term for when fiction refers to fiction
- What did the 8086 (and 8088) do upon encountering an illegal instruction?
- Can an escape pod land on Earth from orbit and not be immediately detected?
- What do I need to do, tax-wise, for a sudden windfall?
- Am I allowed to determine tenets of my contract as a warlock?
- Can you open the door or die? v2
- Getting Ready to Move my Tomato Plants Outside
- What publication claimed that Michael Jackson died in a nuclear holocaust?
- What is the theme of analysis?
- What are the advantages of using TLRs to rangefinders?
- Parallelized for loop in Bash
- Why do the “Shtei HaLechem” not play a prominent part in the davenning for Shavuos?
- What's the reason for the decade jump in the recent X-Men trilogy?
- Was the Lonely Mountain, where Smaug lived, a volcano?

Must a CPU have a GPU if the motherboard provides a display port (when there isn't any separate video card)?

I sent an angry e-mail to my interviewers about a conflict at my home institution. Could this affect my application?

Do Veracrypt encrypted volumes have any kind of brute force protection?

Can Dive Down protect a creature against Pacifism?

How to represent jealousy in a cute way?

Background for black and white chart

Optimising matrix generation time

Manager wants to hire me; HR does not. How to proceed?

Question feed