

## Re: Shell sort for c#

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-08/1119.html>

---

**From:** Mike Kitchen ([publicjoe\\_at\\_hotmail.com](mailto:publicjoe_at_hotmail.com))

**Date:** 08/05/04

Date: Thu, 5 Aug 2004 07:41:49 +0100

Hi Manju

I have a few sort routines on my site that you may wish to look at. These are tried and tested, and ported from C.

They include a shell sort routine which can be found at the following link

<http://www.publicjoe.f9.co.uk/csharp/sort07.html>

Hope this helps

Publicjoe

C# Tutorial at <http://www.publicjoe.f9.co.uk/csharp/tut.html>

C# Snippets at <http://www.publicjoe.f9.co.uk/csharp/snip/snippets.html>

C# Ebook at <http://www.publicjoe.f9.co.uk/csharp/samples/ebook.html>

VB Ebook at <http://www.publicjoe.f9.co.uk/vbnet/samples/ebook.html>

"Majnu" <[spam@majnu.net](mailto:spam@majnu.net)> wrote in message  
news:5906a995.0408041228.4afa3822@posting.google.com...

> *Hi community,*

>

> *just in case somebody needs a shellsort in c#, I rewrote the pascal  
> code that I found in another newsgroup. Here are both. For more  
> explanation on the pascal code you can search for the post on google.*

>

> *I was interested in the shell sort because I needed to sort long lists  
> of data that may already be in order. For that reason I didn't want to  
> use the arraylist's sort method, which uses quick sort (and has a  
> worst case for lists that are already in order). However, I found out  
> that the shell sort performs slower than the arraylist's sort method,  
> even for ordered lists with 1000000 members. I guess this is because  
> the sort method is hard wired into c#, but I'm not sure.*

>

> *Anyway, here it is:*

>

> *public static void shellSort(IList list, int first, int n, IComparer  
> comparer)*

> *{*

> *int dummyRemainder; // not needed but required by System.Math.DivRem*

microsoft.public.dotnet.languages.csharp: Re: Shell sort for c#

```
> int incr = System.Math.DivRem((n * 10), 17, out dummyRemainder);
```