Subject: Re: Funny way how NOT to speedup sorting of small arrays Posted by piotr5 on Sun, 30 Aug 2009 23:38:24 GMT View Forum Message <> Reply to Message

a small update since I wont get the time to make a full test:

with my particular test-data I can only confirm that nothing can drastically outperform a simple sort-loop. (i.e. exchange the minimum with the first element, and continue without that first element.) only through full speed-optimization the divide-and-conquer search for a minimum can really keep up with the loop. especially shell-search isn't much faster than even bubble-sort (all that with sorting only 16 integers)! however, with a trick I managed to find a sorting-algorithm which is about twice as fast: insert-sort. insert-sort in combination with binary search and a mem-copy for the insertion is faster (with my particular test-data) than any of them. it isn't much, but it's noticable. unfortunately memcpy() isn't fair since it doesn't take care of any pick-beaurocracy -- but it certainly is faster than any piece-by-piece swapping or shifting.

in my previous message I said something which is wrong: shell-sort is done by applying insert-sort, and not as I said by recursively applying the same method. recursive shell-sort would only re-do what has already been done in the previous step!

Page 1 of 1 ---- Generated from U++ Forum