How to sort millions of rows of data in a file with less/meagre memory

(From here)

I attended an interview last week and this question was asked:

How do you sort a billion rows of data in a file with only 640KB of memory in a 8080 processor based machine? No virtual memory, no external disk.

I explicitly asked the interviewer if I could use a hard drive, so I can serialize trees as I sort them and then combine at the end. He said no. I tried many ways, different algorithms. Nothing he agreed.

I gave up and asked him politely, "how would you do that?" He bluntly said, "I would not tell you." (The interview ended right after that. I didn't mean to offend him, as a developer, I got curious. Moreover, it was an instinctive question, just as I would ask anyone at my workplace.)

This interview was for a really big bank.

So, how would anyone approach this problem?

algorithm

edited Oct 18 '10 at 16:56
Donotalo
6,088 5 44 79

asked Oct 18 '10 at 16:38
Matty H
56 1 3

10 sounds like he didn't know either!! – Pharabus Oct 18 '10 at 16:42

9 Where do you get the file from if you can't use the drive? It's certainly not going to be held in memory. – Robusto Oct 18 '10 at 16:49

Since the interview was over very quickly, I think maybe you ought to point him here, as some of the best minds in the world can't figure it out either. – KevinDTimm Oct 18 '10 at 16:51

1 The 8080 could address 64K of memory (and 256 I/O ports). What sort of bank switching would be in effect (or did you mean 8086/8088)? – David Thornley Oct 18 '10 at 18:05

1 @Robusto: Possibly that was a question the interviewee was supposed to ask. Asking questions with insufficient information and seeing what the interviewee does about it seems popular this decade. – David Thornley Oct 18 '10 at 19:28

show **3** more comments

9 Answers

Heapsort would be my reccomendation. It's relatively quick when n is large, and you only have to look at three elements with definite indecies at once.

That being said, my intuition tells me that sorting a billion rows on an 8080 even in C would be unfeasibly slow.

edited Oct 18 '10 at 21:08

answered Oct 18 '10 at 16:47
Squirrelsama
1,446 1 11 28

1 +1 if I could ... really, any in-place sort would work, assuming that the "no hard drive" requirement didn't cover the initial data set. Heapsort will be slightly faster than Bubble sort, even on an 8080 :-) – Anon Oct 18 '10 at

17:26

If I had the numbers to back me up here I would, but I guarantee heap sort would be orders of magnitude faster than bubble sort. :D – Squirrelsama Oct 18 '10 at 20:56

1 if you have a different answer, it's quite acceptable in SO to add a second one. I recommend removing your edit and posting heap sort & merge sort as another answer – Tim McNamara Oct 18 '10 at 21:07

Slightly faster? LIGHT YEARS FASTER my friend... – clamchoda Apr 13 '11 at 19:23

add a comment

I would not do it in C#, for starters. Are you sure you have this tagged right? This is a C problem, if it can be solved.

640K only gives you 640 * 1024 * 8 bits so there's no way to solve this as framed. Perhaps that's the answer he/she was looking for. These Investment Bank interviews are something of a mindgame sometimes.

edited Oct 18 '10 at 17:16

answered Oct 18 '10 at 16:43
Steve Townsend
37.9k 3 42 95

+1 for can't do it (or don't do it, either way). – Joel Rondeau Oct 18 '10 at 16:49

1 I agree, it sounds like he might have been asking an "impossible question" to see how the OP responded under pressure. From the way he tells it, he responded exactly appropriately, trying various approaches and then finally giving up with grace. If that isn't good enough for the interviewer... it's probably not going to be a very fun job either, so good riddance. – Ether Oct 18 '10 at 17:14

I don't think there's a C# compiler for the 8080. There were a few C compilers, but the one I had sure didn't meet the C89 standard. – David Thornley Oct 18 '10 at 18:10

add a comment

If speed is not a requirement, you could bubble sort rows in place in the file. This only requires looking at two rows of data at a time, with no external information or storage required.

answered Oct 18 '10 at 16:40
Reed Copsey
332k 27 591 942

@Reed - this involves use of hard drive though, which was ruled out. Possibly questioner got the framing wrong. – Steve Townsend Oct 18 '10 at 16:44

1 I would agree bubble sort, or one of it's derivatives like Cocktail Sort or Comb Sort is the correct answer. – Scott Chamberlain Oct 18 '10 at 16:47

5 If you're using a bubble sort on a billion rows, speed had better *not* be a requirement. :) – Robusto Oct 18 '10 at 16:47

1 Not really, it only requires you read a row into memory, and write a row back into the same location. If you can't read / write to the source file the question is impossible because you don't know what you have to sort... – LorenVS Oct 18 '10 at 16:49

1 @LorenVS: That was my exact thought. This is very different than "serializing trees" as it doesn't require any **extra** hard drive space, as you're sorting in place. Any sort will require changing the info on the hard drive. – Reed Copsey Oct 18 '10 at 16:53

show **1** more comment

Another question to have asked, is "What is the nature of the rows?" If the number of distinct values is low enough, then the answer might be a pigeon hole sort.

For example, say the file to be sorted only contained rows that held a number between 0 and 100 inclusive. Create an array of 101 unsigned 32 bit or 64 bit integers with a value of 0. As you read a row, use it to index the array and increament the count of that element. Once the file is read, start at 0, read the the number of zeros read and spit out that many, go to 1, repeat. Expand the array size as needed to handle the set of numbers coming through. Of course there are limits, say the values that can be seen span from -2e9 to +2e9. That's going to require 4e9 bins, which is not going to fit in 640K of RAM.

If instead the rows are strings, but you are still looking at a small enough set of distinct value, then use an associative array or hash table to hold the counts.

answered Oct 18 '10 at 21:23
Shannon Severance
8,576 1 15 42

add a comment

Knuth has a whole section on external sorting; this was commonplace back when there were no hard drives & not much memory, and tape drives were the norm. Look at the wikipedia page, and/or vol. 3 of Knuth's Art of Computer Programming.

I agree with Robusto's comment:

> Where do you get the file from if you can't use the drive? It's certainly not going to be held in memory.

Not enough problem definition.

answered Oct 18 '10 at 18:06
Jason S
64.3k 64 312 572

I should have asked him that question. Where is file located if no external drive? It never occurred to me in the interview. Anyway, it was a C# position, and the interview was in Java. I kept on taking him back to C# world, he kept on insisting on Java. (I worked in Java 5 years ago and it was on Resume, not to be unfair to the interviewer, I couldn't say I don't know Java, which is partly correct, since its been long). – Matty H Oct 18 '10 at 20:35

add a comment

The more I think about this, the more I think merge sort would work very well within the memory window we're given.

Let's say you have x memory available. Divide the billion entries into billion/x + 1 sections and heapsort them (heapsort because no extra memory is required and it's O(2n(log n)) time). When all sections are heapsorted, do a merge sort starting across the first elements of all sections. This will work so long as you have more than sqrt(billion) memory to work with given basic 8080 OS memory usage.

Doing the math, this assumes that each data row is less than 165 bits.

edited Oct 18 '10 at 21:15

answered Oct 18 '10 at 21:08
Squirrelsama
1,446 1 11 28

add a comment

Obviously you have to be able to read and write to the billion row file. The constraint of no external disk means you must restrict yourself to in-place algorithms or make some assumptions about the starting conditions and distribution of data so that you can keep the data sorted as it is added to the file (e.g. use the key as the index and create a large enough file to hold the expected number of keys).

If you must start with an unsorted file and sort it, you can use merge an in-place merge sort operating on very small chunks of the file. Since no constraints are made on the access times of the storage media, it may be very fast.

answered Oct 18 '10 at 21:24
Justin
2,481 1 16 43

2 I think this should be the top answer, was about to post something very similar. Even if the list is on a tape reel, you can always read, sort and write subsets of the list, providing you have enough memory to hold at least 2 rows. – jambox Dec 29 '10 at 10:50

add a comment

I'd use the GPU! Even on a fast computer, the GPU is often faster at sorting. And I don't know how big the "rows" are, but it's not hard to find 1GB video cards, so that answers the storage question, too.

Besides, if I had to work on an 8080, I'd definitely want to put the sweetest graphics card I could find on there.

You just have to be ready for the follow-up question: "How do you get an 8080 to talk to a modern PCI Express 2.0 x16 card?". I have discovered a truly marvelous method, but this textarea is too narrow to contain it.

answered Oct 18 '10 at 17:39

Ken

1,924 2 5 12

2 Ha ha. +1 for creativity. While you're at it, hook the PCI card up to a Cray. – LarsH Oct 18 '10 at 18:42

add a comment

You can find the discussion on a similar problem in Jon Bentley *Programming Pearls* Column. 1. Here Bentley deals with a problem of sorting millions of area codes which are guaranteed to be unique by using a bitset data-structure.

answered Aug 1 '11 at 9:03

vine'th

2,339 2 11 16

add a comment

Not the answer you're looking for? Browse other questions tagged algorithm or ask your own question.