

# Time Trials for Lertap 5.10.7 (Windows)

Larry Nelson, Curtin University  
Western Australia

The last update of this document was on: 1 February 2016

---

Computers: **HP** Spectre x360, Intel i5-6200U, 8GB / 256GB SDD  
Sony **Vaio** Pro 13, i5-4200U, 4GB / 128GB SDD  
MacBookPro (**MBP**) Late 2013, Core i5, 8GB / 128GB SDD

Software: On the **HP**:  
Windows 10 Home; Excel 2010 (14.0.7165.5000, 64-bit);  
Excel 2016 (16.0.6326.1010, 64-bit).  
On the **Vaio**:  
Windows 8.1; Excel 2013 (15.0.4779.1001, 64-bit).  
On the **MBP**:  
OSX 10.11.12; Excel 2011 (14.5.9, 151119).

Notes: The HP and the Vaio are laptop computers designed to provide good battery life, something they do at the expense of processor speed. Note they are running 64-bit versions of Excel. *The 32-bit versions of Excel 2013 and 2016 are quite likely to run into memory restrictions when Lertap 5 is run with N=5,000 or more.*

Times below are shown in minutes:seconds. A time of 1:02, for example, means 1 minute 2 seconds. N is number of students. Subs is number of subtests in use.

The datasets may be found at, and downloaded from, [this site](#).

No.	Name	N	Items	Subs	Excel 2010	Excel 2013	Excel 2016
1	Cook's Tour	60	35	3	0:07	0:10	0:13
2	HalfTime	424	100	1	0:15	0:27	0:32
3	MNursing	1,769	60	1	0:18	0:46	0:52
4	Zmed	2,470	100	1	0:27	0:57	1:02
5	LenguaBIg	5,504	50	1	0:36	2:31	2:44
6	LenguaBIg	5,504	50	3	1:33	7:57	8:09

Two larger datasets were also put through time trials. With N=35,000 items=44 and subs=1, times with Excel 2010, 2013, and 2016 were: 3:16, 15:10, 15:45. The same test with N=70,000 gave these times: 6:53, 30:25, 32:11.

On the **MacBookPro** with Excel 2011 and Lertap 5.9.2.2 (version of 10 January 2016), the Cook's Tour took 1:49; HalfTime 5:49; MNursing 6:30; Zmed 9:10, and LenguaBIg (subs=1) 13:25.

## **Production mode**

The figures above were derived by using a stopwatch to time Lertap 5.10.7 as it ran in "[Production mode](#)". We put "yes" in row 35 of the System worksheet, and "no" in rows 36, 37, and 38. The result was equivalent to running the "[Interpret](#)" option followed immediately by the "[Elmillion](#)" option.

A check was made by running through each dataset twice.

## **Upgrade?**

The times shown in the table result from running version 5.10.7 of Lertap, a version released in early January 2016.

This version specifically targeted processing speed. Sections of Lertap's computer code were revised in order to get better throughput, and, basically, there have been marked improvements. For example, under the previous version running with Excel 2013, HalfTime took 4:15, MNursing 8:50, LenguaBIg (with subs=3) 30:45.

Users with earlier versions of Lertap should contact [support@lertap.com](mailto:support@lertap.com) if they have an interest in downloading version 5.10.7.

## **Comments**

It is likely that users with more powerful computers, such as desktop units running with an i7 core, will observe better speed. However, in 2015 we compared the Sony Vaio used here with an i7-based Microsoft Surface Pro 3. No time differences were noted at that time when using Excel 2013 on both machines to process the HalfTime and MNursing datasets.

Out of interest, we also tried version 5.10.7 with an HP Stream 8 tablet running Windows 10 and 32-bit Excel 2016. The Stream had only 1GB of RAM and used an Intel Atom Z3735G processor. HalfTime was processed in 1:30 and MNursing in 2:23. This was with the 32-bit version of Excel 2016. These times are relatively slow, but nonetheless Lertap's output was not altered in any way – its graphs, including item [response plots](#), rendered very well; even the extraction of [eigenvalues](#), a computationally intense process, worked fine. (With Excel 2010 on the same tablet, HalfTime took 0:48 and MNursing 1:02.)

Users should note the comments made above regarding 32-bit versions of Excel 2013 and 2016: they're subject to memory management limitations which are very likely to impede the application of Lertap when N exceeds 5,000. This is true even on computers with 8GB and more of memory; 32-bit Excel 2013 and 2016 are constrained to operate in just 2GB of memory. Refer to [this webpage](#) for related comments.

## **Macs and Excel and Lertap 5**

There is now a version of Excel 2016 for Macintosh computers. However it does not yet work with Lertap. It looks ever so much like the Windows product, but its version of VBA, the programming language used in Excel and other Office products, is not yet as thoroughly developed.

Some of the code changes behind version 5.10.7 have now been incorporated in the Excel 2011 version of Lertap for the Mac. While the times reported above for the Excel

2011 version are large, they are nonetheless well under what they were before. For example, with the previous Excel 2011 version of Lertap it took over 45 minutes to process MNursing.

### **How to run faster** (without buying new jiggers)

One of the slower parts of running Lertap is related to the creation of the "[Stats\\_ul](#)" reports. It is possible to turn the Stats\_ul reports off – this is done by putting "no" in row 10 of the [System](#) worksheet.

Another time consuming activity concerns getting Excel to adjust its page breaks so that quantile / [quintile plots](#) do not get spilt in the middle when they're printed. Adjusting page breaks may be turned off by putting "no" in row 92 of the [System](#) worksheet.

Of the three versions of Excel used for our time trials, Excel 2010 was clearly the fastest. This version may still be purchased, it's still on the market. And, it would seem likely that IT departments may have Excel 2010 installation discs in their archives. It's possible to have more than one version of Excel installed on Windows computers. (Our computers each have at least two versions; one of them is always Excel 2010.)

### **Updates Summary**

Visit [this document](#) to track Lertap 5 developments. Also, see [this webpage](#).