

## The Computer Language Benchmarks Game

### mandelbrot

#### description

Always look at the source code.

Look at the slower simple sequential programs, *and* look at the parallel programs written for multicore, *and* look at the manually vectorized SIMD programs.

×	source	secs	mem	gz	busy	cpu load
1.0	<b>C++ g++ #4</b>	<b>0.84</b>	34,780	3542	3.27	99% 98% 98% 96%
1.1	<u>C++ g++</u>	0.89	32,592	1791	3.48	97% 98% 97% 99%
1.1	<b>Rust #8</b>	<b>0.93</b>	32,676	763	3.70	100% 99% 100% 100%
1.1	<u>Rust #7</u>	0.94	32,468	757	3.72	99% 99% 100% 99%
1.2	<u>C++ g++ #6</u>	0.97	32,876	1002	3.83	100% 99% 99% 99%
1.4	<u>Rust #5</u>	1.20	34,332	719	4.71	98% 98% 98% 99%
1.5	<u>C gcc #6</u>	<b>1.27</b>	31,792	1135	5.08	100% 100% 99% 100%
1.6	<b>Julia #8</b>	<b>1.32</b>	227,836	624	4.66	98% 86% 85% 85%
1.6	<b>Swift #7</b>	<b>1.34</b>	41,120	1140	5.34	99% 99% 100% 100%
1.6	<u>Swift #6</u>	1.35	42,752	1141	5.36	99% 99% 99% 100%
1.6	<u>Rust #6</u>	1.36	33,384	1332	5.37	100% 99% 99% 99%
1.7	<u>Intel Fortran #8</u>	<b>1.42</b>	36,184	957	5.63	99% 100% 99% 99%
1.8	<u>Rust #3</u>	1.48	14,276	1007	5.91	100% 99% 99% 100%
1.8	<b>Haskell GHC #3</b>	<b>1.51</b>	37,504	1975	5.98	99% 99% 99% 99%
2.2	<u>Julia #7</u>	1.81	228,368	619	6.67	90% 90% 90% 99%
2.2	<u>Julia #5</u>	1.84	271,872	571	6.85	91% 91% 99% 91%
2.5	<u>Julia #3</u>	2.11	267,304	579	7.86	91% 91% 90% 100%
2.8	<u>Julia #6</u>	2.36	258,096	562	6.80	49% 57% 92% 91%
3.3	<u>Julia #4</u>	2.74	263,472	574	7.74	46% 55% 91% 91%
3.6	<u>Rust</u>	3.04	39,092	868	11.93	98% 98% 99% 98%
3.8	<b>C# .NET</b>	<b>3.14</b>	64,940	1974	12.30	99% 98% 97% 97%
3.8	<u>C# .NET #9</u>	3.14	64,736	816	12.29	97% 99% 97% 97%

3.8	<u>Intel Fortran #6</u>	3.16	75,852	661	10.64	79%	79%	79%	100%
3.8	<u>C gcc #8</u>	3.17	33,084	782	12.15	100%	94%	96%	94%
4.0	<b>Chapel #3</b>	<b>3.34</b>	36,104	588	13.22	99%	99%	99%	99%
4.1	<u>C gcc #4</u>	3.46	32,552	799	13.82	100%	100%	100%	100%
4.1	<u>C++ g++ #7</u>	3.46	34,776	1017	13.81	100%	100%	100%	99%
4.2	<u>C gcc #7</u>	3.48	33,044	994	13.86	99%	100%	99%	100%
4.2	<u>C gcc</u>	3.49	29,024	822	13.94	100%	100%	100%	100%
4.2	<u>C++ g++ #9</u>	3.50	34,452	726	13.50	95%	95%	99%	96%
4.2	<u>C gcc #3</u>	3.51	32,400	763	14.00	100%	99%	99%	100%
4.5	<b>Go #4</b>	<b>3.73</b>	34,192	905	14.85	99%	99%	99%	100%
4.5	<u>Go #3</u>	3.75	35,204	894	14.97	100%	100%	100%	100%
4.5	<b>F# .NET #6</b>	<b>3.77</b>	64,912	933	14.73	99%	97%	98%	97%
4.6	<u>Free Pascal #8</u>	<b>3.86</b>	31,404	974	15.38	99%	99%	100%	99%
4.8	<b>Ada 2012 GNAT #3</b>	<b>4.02</b>	35,812	1819	15.96	100%	99%	99%	100%
4.8	<u>F# .NET #5</u>	4.02	65,040	897	15.76	98%	98%	97%	99%
4.8	<u>F# .NET</u>	4.04	65,208	877	15.81	98%	97%	98%	99%
4.8	<b>Node js #3</b>	<b>4.04</b>	95,888	1122	15.91	100%	98%	98%	99%
4.9	<b>Lisp SBCL #4</b>	<b>4.09</b>	46,684	2447	16.19	99%	99%	99%	99%
4.9	<b>Java #2</b>	<b>4.12</b>	70,952	796	16.22	98%	98%	98%	99%
5.0	<u>C# .NET #5</u>	4.14	64,736	839	16.24	98%	99%	98%	98%
5.0	<u>C gcc #9</u>	4.16	33,116	694	16.05	95%	100%	95%	95%
5.1	<u>Java #6</u>	4.30	69,004	802	16.91	99%	99%	98%	98%
5.3	<u>Java #4</u>	4.42	67,580	660	16.17	96%	84%	87%	100%
5.5	<u>Lisp SBCL</u>	4.58	46,644	2473	16.61	79%	99%	99%	85%
5.7	<u>C++ g++ #8</u>	4.81	34,424	742	13.31	39%	100%	38%	100%
6.8	<u>Go</u>	5.72	33,968	823	22.83	100%	100%	100%	100%
7.8	<u>Free Pascal #7</u>	6.55	36,696	950	26.12	100%	100%	100%	100%
7.9	<u>Haskell GHC #2</u>	6.60	39,668	782	26.33	100%	100%	100%	100%
8.1	<u>Chapel</u>	6.77	35,928	460	26.90	100%	99%	99%	99%
8.1	<u>Go #2</u>	6.79	33,240	837	27.11	100%	100%	100%	100%
8.2	<u>Go #6</u>	6.85	32,816	700	27.29	100%	100%	100%	100%
8.8	<u>Swift #5</u>	7.33	42,920	659	27.07	90%	100%	90%	90%

8.8	<u>Java #3</u>	7.35	70,360	903	29.08	99%	99%	99%	99%
8.9	<u>Free Pascal #6</u>	7.45	35,876	802	29.77	100%	100%	100%	100%
9.0	<b>OCaml</b>	<b>7.53</b>	4,480	717	30.02	100%	100%	100%	100%
10	<u>Dart #3</u>	<b>8.72</b>	57,164	957	33.81	95%	100%	99%	95%
12	<u>Intel Fortran #4</u>	10.36	75,752	618	27.09	91%	91%	33%	46%
12	<b>Racket #4</b>	<b>10.44</b>	126,440	801	41.22	98%	98%	98%	100%
13	<u>Free Pascal</u>	10.96	32,992	846	29.48	94%	94%	44%	38%
13	<u>Free Pascal #2</u>	10.97	32,992	846	29.50	94%	44%	38%	94%
13	<u>Lisp SBCL #3</u>	11.08	50,352	894	29.66	34%	99%	100%	34%
14	<u>Dart #2</u>	11.80	88,664	907	31.85	35%	100%	36%	99%
16	<u>C++ g++ #5</u>	13.36	33,504	590	13.38	0%	0%	100%	0%
21	<u>C++ g++ #2</u>	17.46	34,164	692	17.51	0%	100%	0%	0%
30	<b>PHP #3</b>	<b>24.89</b>	135,480	875	99.12	100%	100%	99%	100%
30	<u>C gcc #2</u>	25.41	764	400	25.66	0%	0%	1%	100%
31	<u>Swift</u>	26.29	7,552	394	26.54	1%	0%	0%	100%
32	<u>Free Pascal #3</u>	26.52	8	526	26.74	0%	0%	100%	1%
32	<u>Julia</u>	26.73	203,668	373	27.41	100%	1%	1%	1%
33	<u>Java</u>	27.79	35,568	665	27.91	0%	100%	0%	0%
34	<u>OCaml #6</u>	28.22	3,096	444	28.32	0%	0%	0%	100%
34	<u>Ada 2012 GNAT #2</u>	28.42	1,984	577	28.73	1%	0%	100%	0%
34	<u>C++ g++ #3</u>	28.60	1,784	414	28.87	1%	0%	100%	0%
42	<u>Racket #3</u>	34.76	66,608	802	35.01	1%	0%	100%	0%
53	<u>Intel Fortran #5</u>	44.57	44,056	520	46.37	1%	2%	1%	100%
53	<b>Erlang #3</b>	<b>44.66</b>	67,152	792	178.29	100%	100%	100%	100%
54	<u>Erlang #4</u>	45.07	64,896	806	179.91	100%	100%	100%	100%
55	<u>Erlang #2</u>	46.05	765,680	613	182.90	99%	99%	99%	99%
58	<u>Erlang</u>	48.67	1,649,804	534	188.79	97%	97%	97%	97%
59	<u>Racket</u>	49.38	68,144	517	49.56	0%	100%	0%	0%
61	<u>Racket #2</u>	51.37	67,020	585	51.84	0%	1%	0%	100%
105	<u>PHP</u>	87.65	12,176	449	93.17	1%	2%	3%	100%
123	<u>VW Smalltalk #2</u>	<b>102.47</b>	177,424	929	5 min	85%	93%	88%	77%
135	<b>Lua #6</b>	<b>113.06</b>	18,572	623	6 min	86%	100%	77%	86%

195	<b><u>Python 3 #7</u></b>	<b>163.32</b>	12,080	688	10 min	98%	98%	98%	98%
283	<u>Lua #3</u>	236.48	2,584	358	238.72	100%	0%	0%	0%
289	<u>Lua</u>	241.77	2,564	359	243.65	0%	0%	0%	100%
314	<b><u>Ruby #2</u></b>	<b>262.79</b>	69,328	880	17 min	100%	100%	100%	100%
388	<u>VW Smalltalk</u>	5 min	54,968	603	5 min	0%	100%	0%	0%
445	<u>Lua #2</u>	6 min	1,236	367	6 min	0%	0%	0%	100%
464	<u>Ruby #5</u>	6 min	59,680	970	25 min	100%	100%	100%	100%
649	<b><u>Perl</u></b>	<b>9 min</b>	45,036	586	36 min	100%	100%	100%	100%
	<u>Ruby #4</u>	11 min	80,868	846	11 min	25%	25%	25%	25%
	<u>Ruby #3</u>	26 min	23,360	307	26 min	0%	0%	0%	100%
	<u>Matz's Ruby #5</u>	<b>32 min</b>	53,240	970	2h 10 min	100%	100%	100%	100%
	<u>Matz's Ruby #2</u>	39 min	64,428	880	2h 37 min	100%	100%	100%	100%
	<u>Matz's Ruby #6</u>	39 min	131,576	918	2h 32 min	96%	96%	95%	95%
	<u>Python 3 #5</u>	Failed							
	<u>Python 3 #2</u>	Timed Out							
	<u>Ruby #7</u>	Failed							
	<u>Ruby #6</u>	Failed							
	<u>Matz's Ruby #7</u>	Failed							
	<u>Matz's Ruby #4</u>	Timed Out							
	<u>Matz's Ruby #3</u>	Timed Out							
	<u>Swift #3</u>	Bad Output							

**by secs**

by mem

by gz

by busy

## How programs are measured