

The Computer Language Benchmarks Game

binary-trees

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	C++ g++ #7	0.94	176,428	1122	3.39	86%	88%	100%	85%
1.2	Rust #5	1.09	198,728	765	3.90	87%	98%	88%	86%
1.2	<u>C++ g++ #5</u>	1.14	200,396	885	4.17	100%	89%	87%	89%
1.3	<u>Rust #2</u>	1.19	200,112	721	4.32	88%	100%	88%	88%
1.3	<u>C++ g++ #4</u>	1.25	134,416	1270	3.64	74%	61%	69%	88%
1.3	<u>Rust #4</u>	1.27	164,332	756	4.64	91%	88%	88%	99%
1.4	<u>Rust #3</u>	1.36	169,516	754	4.92	87%	88%	98%	90%
1.6	<u>C++ g++ #9</u>	1.51	125,080	809	4.21	100%	69%	56%	54%
1.6	C gcc #2	1.54	168,832	809	4.35	60%	67%	57%	100%
1.9	<u>C gcc #3</u>	1.79	119,812	836	4.82	53%	52%	99%	66%
1.9	<u>C++ g++ #3</u>	1.83	265,396	945	4.59	40%	58%	88%	64%
2.0	<u>C++ g++ #6</u>	1.84	265,756	844	4.55	49%	40%	88%	70%
2.1	<u>Free Pascal #6</u>	1.99	132,728	959	4.97	100%	58%	50%	42%
2.1	<u>Free Pascal #7</u>	2.01	132,732	1003	5.00	100%	58%	42%	49%
2.2	<u>Free Pascal #5</u>	2.03	132,728	953	4.72	99%	49%	43%	42%
2.2	<u>C++ g++ #8</u>	2.10	265,768	820	5.15	44%	48%	88%	66%
2.3	<u>Rust</u>	2.21	133,332	751	7.84	89%	93%	84%	88%
2.5	<u>Intel Fortran #2</u>	2.38	133,752	1148	7.45	100%	70%	70%	73%
2.6	Java #7	2.48	1,725,776	835	7.86	74%	75%	97%	72%
2.8	<u>C++ g++</u>	2.60	1,954,096	754	8.99	87%	83%	92%	84%

3.0	<u>Lisp SBCL #3</u>	2.84	529,972	932	7.09	73%	46%	63%	67%
3.3	<u>Ada 2012 GNAT #4</u>	3.12	202,368	2107	9.63	79%	93%	67%	71%
4.9	<u>Java #3</u>	4.57	2,091,868	540	5.49	10%	9%	93%	9%
4.9	<u>Java #6</u>	4.59	2,066,412	529	5.43	23%	10%	76%	9%
5.0	<u>OCaml #2</u>	4.68	152,716	751	13.23	78%	99%	42%	64%
5.1	<u>Java #2</u>	4.77	2,091,580	552	5.62	17%	30%	20%	51%
5.1	<u>Racket #4</u>	4.81	555,172	853	9.55	47%	46%	37%	69%
5.1	<u>Java #4</u>	4.81	2,110,392	840	5.67	91%	6%	7%	13%
5.1	<u>C# .NET #3</u>	4.81	1,881,564	676	15.47	81%	74%	83%	83%
5.2	<u>C# .NET #2</u>	4.89	1,617,228	705	15.77	74%	91%	79%	79%
5.2	<u>C# .NET #7</u>	4.91	1,583,688	694	15.97	76%	92%	77%	81%
5.3	<u>Haskell GHC #7</u>	5.02	449,868	811	18.05	90%	85%	93%	91%
5.3	<u>C# .NET</u>	5.03	1,910,092	657	16.35	83%	79%	84%	79%
5.5	<u>Erlang #2</u>	5.19	419,540	478	15.94	71%	74%	81%	81%
5.5	<u>Haskell GHC #6</u>	5.21	457,732	592	18.75	89%	87%	97%	88%
5.7	<u>Haskell GHC #8</u>	5.32	355,116	799	18.85	93%	89%	88%	85%
6.4	<u>F# .NET #7</u>	6.06	1,469,616	516	16.18	63%	69%	62%	72%
6.5	<u>Racket #3</u>	6.08	317,712	473	6.21	1%	100%	0%	0%
7.6	<u>Haskell GHC #5</u>	7.13	699,576	561	25.82	88%	88%	98%	88%
7.6	<u>Node js #6</u>	7.20	1,282,116	744	20.39	83%	70%	68%	63%
7.7	<u>Haskell GHC #3</u>	7.21	689,468	538	26.92	90%	98%	92%	93%
7.9	<u>Julia #3</u>	7.43	482,620	634	23.65	69%	100%	78%	72%
8.5	<u>Chapel #3</u>	7.96	298,772	488	24.53	99%	74%	76%	58%
9.1	<u>C gcc #5</u>	8.55	351,376	908	31.51	92%	93%	93%	91%
9.7	<u>Racket #2</u>	9.12	318,372	416	9.33	1%	100%	0%	1%
10	<u>Lisp SBCL</u>	9.78	552,028	589	10.02	0%	100%	1%	1%
11	<u>Dart #7</u>	9.90	539,308	863	30.78	83%	74%	70%	85%
11	<u>Dart #4</u>	10.03	510,088	1387	31.01	69%	79%	93%	68%
11	<u>Dart #5</u>	10.24	443,504	1016	28.25	61%	62%	86%	68%

11	<u>OCaml #5</u>	10.54	154,276	461	10.67	100%	1%	0%	0%
12	<u>Node js</u>	11.40	1,048,328	711	33.03	72%	73%	73%	72%
12	<u>Erlang</u>	11.74	446,540	420	12.05	6%	88%	1%	7%
13	<u>Ada 2012 GNAT #3</u>	11.88	528,072	1266	33.61	68%	75%	70%	71%
13	<u>Go #8</u>	12.23	392,744	1017	41.62	83%	87%	85%	85%
13	<u>Go #5</u>	12.48	342,520	950	49.63	99%	99%	99%	99%
14	<u>Go #3</u>	12.93	364,716	799	50.32	99%	96%	97%	97%
15	<u>Haskell GHC</u>	14.10	667,060	494	26.89	30%	30%	30%	100%
15	<u>Haskell GHC #2</u>	14.41	454,324	491	25.94	30%	100%	30%	21%
15	<u>Dart #3</u>	14.53	980,520	1212	49.72	92%	85%	80%	85%
16	<u>Dart</u>	15.28	405,784	482	22.81	5%	94%	36%	13%
16	<u>Racket</u>	15.50	436,996	476	16.48	3%	100%	3%	0%
17	<u>Swift #3</u>	16.15	699,872	862	51.81	77%	66%	79%	100%
17	<u>Swift #2</u>	16.26	699,876	816	51.95	100%	67%	78%	75%
17	<u>VW Smalltalk #3</u>	16.33	375,880	930	37.41	70%	46%	70%	43%
18	<u>Swift #5</u>	16.72	732,888	806	50.18	67%	95%	65%	73%
19	<u>Julia #4</u>	17.43	418,700	372	45.08	57%	66%	74%	61%
20	<u>PHP #7</u>	18.64	1,588,704	760	67.29	89%	90%	96%	86%
20	<u>Swift #4</u>	18.77	699,884	765	59.80	75%	66%	77%	100%
20	<u>Julia #2</u>	19.19	479,728	423	36.92	30%	32%	39%	92%
21	<u>Swift</u>	19.39	733,308	714	58.01	72%	95%	64%	68%
23	<u>PHP #4</u>	21.40	1,558,524	779	64.07	56%	85%	86%	73%
23	<u>C gcc</u>	22.10	263,568	654	22.40	100%	0%	0%	1%
24	<u>Ada 2012 GNAT</u>	22.40	265,456	888	22.69	0%	1%	100%	0%
24	<u>C++ g++ #2</u>	22.47	265,172	508	22.72	0%	0%	100%	1%
25	<u>Ruby #5</u>	23.80	566,560	1008	67.07	58%	96%	63%	64%
30	<u>PHP #6</u>	27.89	804,252	868	92.49	78%	78%	98%	78%
30	<u>Go #7</u>	27.98	260,540	525	50.84	42%	47%	44%	48%
30	<u>Go</u>	28.00	250,560	482	49.13	41%	51%	43%	40%

30	<u>Go #6</u>	28.41	407,668	611	29.60	98%	3%	1%	2%
30	<u>Julia #5</u>	28.54	686,764	390	77.14	56%	70%	66%	77%
30	<u>Intel Fortran</u>	28.71	525,624	773	32.74	4%	100%	5%	5%
35	<u>Node js #7</u>	32.69	1,347,080	451	38.11	18%	42%	30%	26%
36	<u>PHP #5</u>	33.78	777,220	1040	88.44	75%	80%	47%	60%
41	<u>VW Smalltalk</u>	38.24	375,680	711	38.56	0%	0%	100%	0%
51	<u>Python 3 #4</u>	48.03	462,732	472	174.44	89%	97%	88%	89%
51	<u>Lua #4</u>	48.15	923,628	664	178.89	97%	92%	91%	91%
54	<u>Python 3</u>	51.07	278,324	589	178.28	85%	86%	93%	85%
56	<u>Perl #5</u>	53.02	1,048,568	797	195.75	91%	88%	100%	90%
60	<u>Ruby #4</u>	56.61	562,776	364	59.20	0%	1%	4%	100%
62	<u>Ruby</u>	58.17	562,756	376	60.68	2%	2%	4%	96%
62	<u>Ruby #3</u>	58.41	862,008	400	60.56	1%	0%	3%	100%
62	<u>Ruby #2</u>	58.53	562,780	378	61.83	0%	4%	100%	1%
67	<u>PHP</u>	63.06	747,960	451	67.59	2%	2%	3%	100%
68	<u>PHP #2</u>	64.14	747,932	423	68.01	4%	2%	97%	2%
80	<u>Python 3 #3</u>	75.49	677,620	1238	220.59	66%	69%	66%	92%
85	<u>Perl #4</u>	79.85	2,519,676	761	291.24	98%	91%	90%	86%
88	<u>Lua #3</u>	82.85	923,584	684	165.86	44%	50%	87%	20%
121	<u>Perl #3</u>	114.27	2,200,624	648	6 min	97%	74%	90%	78%
157	<u>Lua #2</u>	147.61	947,952	412	148.92	0%	1%	0%	100%
157	<u>Python 3 #2</u>	148.16	274,244	338	149.34	0%	0%	0%	100%
268	<u>Perl</u>	252.06	599,456	395	254.71	100%	1%	0%	0%
336	<u>Matz's Ruby #5</u>	5 min	1,413,456	1008	16 min	73%	72%	75%	95%
603	<u>Matz's Ruby #3</u>	9 min	840,244	400	9 min	0%	3%	0%	97%
731	<u>Matz's Ruby #4</u>	11 min	1,560,732	364	12 min	15%	62%	2%	27%
	<u>Matz's Ruby</u>	12 min	1,559,732	376	12 min	0%	100%	0%	0%
	<u>Matz's Ruby #2</u>	12 min	1,558,884	378	12 min	49%	0%	51%	0%
	<u>PHP #3</u>	Failed							

Swift #9

Make Error

by secs

by mem

by gz

by busy

How programs are measured