

## Algorytm 2

### Opis algorytmu

Pomysł polega na tym, że dzielimy dane ze względu na przedział do którego należy wartość, które zapisujemy do kubelków. Każdy wątek analizuje część tablicy i szuka odpowiedniego kubelka. Zapamiętujemy liczbę wartości w każdym kubelku. Następnie sortujemy każde wartości w kubelkach algorytmem szybkiego sortowania. Ponieważ zapamiętaliśmy ile jest liczb w każdym kubelku, możemy po kolei wpisywać wartości z kubelków do ostatecznego wyniku w sposób liniowy względem liczby kubelków. Dzięki temu możemy równoległe wykonywać sortowanie szybkie osiągając lepsze rezultaty niż sekwencyjny algorytm,

### Pytanie 1

*Czy potrzebna jest jakaś ochrona danych wspólnych (tablica początkowa: przy odczycie i przy zapisie; kubelki: przy zapisie, sortowaniu kubelka, odczycie)?*

Nie ma potrzeby ochrony tablicy początkowej przy odczycie - wartości w zależności od spełnienia warunku zostają zapisane do nowego miejsca w pamięci (kubelka). Natomiast warto uwzględnić pewną różnorodność w sposobie przeszukiwania tablicy, ponieważ odnośnienie się do tej samej wartości przez różne wątki może być niewydajne. Kubelki przy zapisie i sortowaniu danych z tablicy również nie potrzebują ochrony, ponieważ każdy wątek operuje na pojedynczym kubelku. Łączenie kubelków jest sekwencyjne, więc również nie wymaga ochrony,

### Pytanie 2

*Jaki jest rząd złożoności obliczeniowej algorytmu, a jaka jest praca algorytmu równoległego, czy algorytm jest sekwencyjnie-efektywny?*

Przygotowanie kubelków:  $O(n \cdot \text{buckets}) \sim O(n)$

Podział na kubelki:  $O(n)$

Sortowanie kubelków:  $O(n \cdot \log n / \text{buckets}) \sim O(n \cdot \log n)$

Łączenie kubelków:  $O(\text{buckets}) \sim O(1)$

Złożoność całego algorytmu:  $O(n \cdot \log n)$

Zauważmy, że głównym parametrem wpływającym na efektywność algorytmu będzie tutaj rozłożenie danych. Jeżeli dane będą rozłożone równomiernie, będziemy mogli efektywnie podzielić dane do sortowania, dzięki czemu redukujemy złożoność tego procesu przy wzroście liczby kubelków (zakładając, że liczba wątków jest proporcjonalna do liczby kubelków). A więc algorytm jest sekwencyjnie-efektywny, natomiast dla niewygodnych danych nie osiągniemy oczekiwanego wyniku. Zauważyć należy również, że przy wzroście liczby kubelków (wątków) znacznie rośnie złożoność przygotowania kubelków oraz łączenie kubelków (liniowo), natomiast sortowanie przy dużej liczbie kubelków przestaje mieć znaczenie. Dlatego napotkamy na moment, gdy liczba wątków/kubelków osiągnie taką liczbę, że narzuty będą powodowały obniżenie efektywności algorytmu.

### Komputer

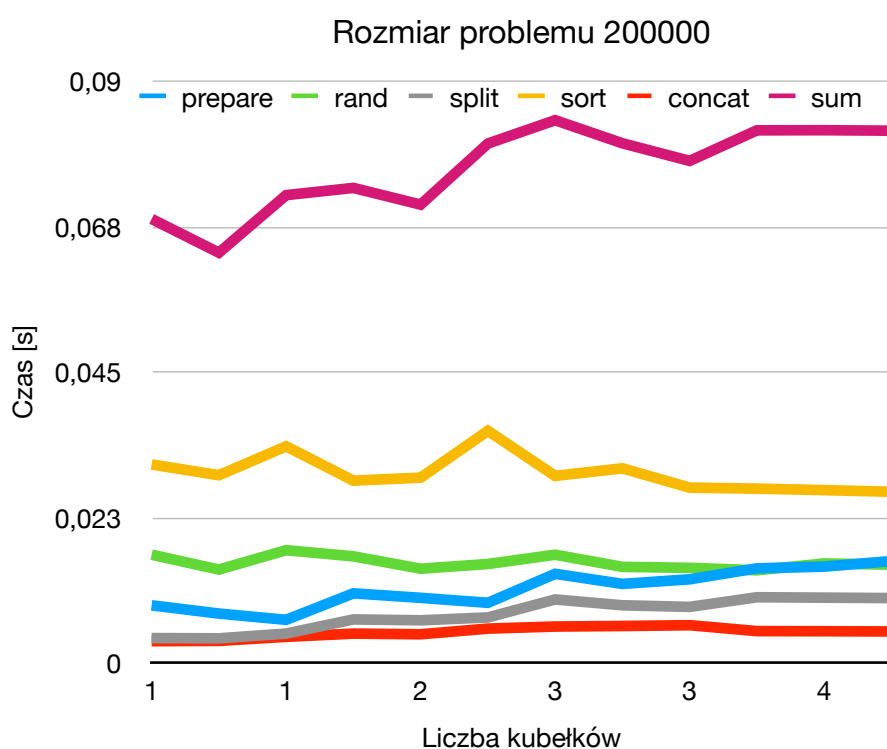
Testy przeprowadzałem na własnym urządzeniu wyposażonym w 8 rdzeniowy/16 wątkowy procesor wyposażony w 16MB pamięci cache L3, 16GB pamięci RAM oraz system operacyjny MacOS 11. Korzystałem z kompilatora clang++ bez optymalizacji.

### Metodologia

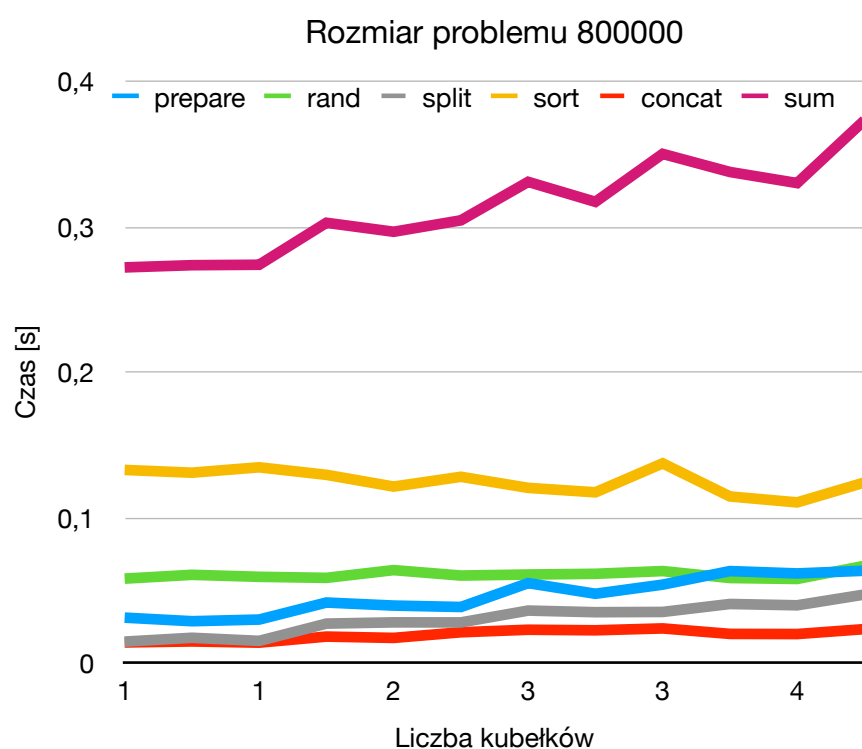
Każdy test przeprowadziłem 3 razy oraz wyciągałem średnią. Wszystkie wartości są w sekundach. Każdy pomiar trwał krócej niż sekundę.

### Algorytm sekwencyjny

Definiując jeden wątek manipulowałem rozmiarem problemu oraz liczbą kubelków w celu uzyskania najoptymalniejszej proporcji rozmiar problemu do liczby kubelków dla każdego wątku. Bez zaskoczenia najlepiej sprawdza się jeden kubek per wątek. Było to oczekiwane, ponieważ obsługa każdego kubelka wprowadza narzuty, które nie zostaną "nadgonione" przez wielowątkowość, ponieważ jeden wątek powinien obsługiwać jedno uzupełnianie kubelka na raz. Zatem na tym etapie reszta testów będzie przeprowadzana w taki sposób, aby dla jednego wątku przypadał jeden kubek.



Algorytm sekwencyjny 2

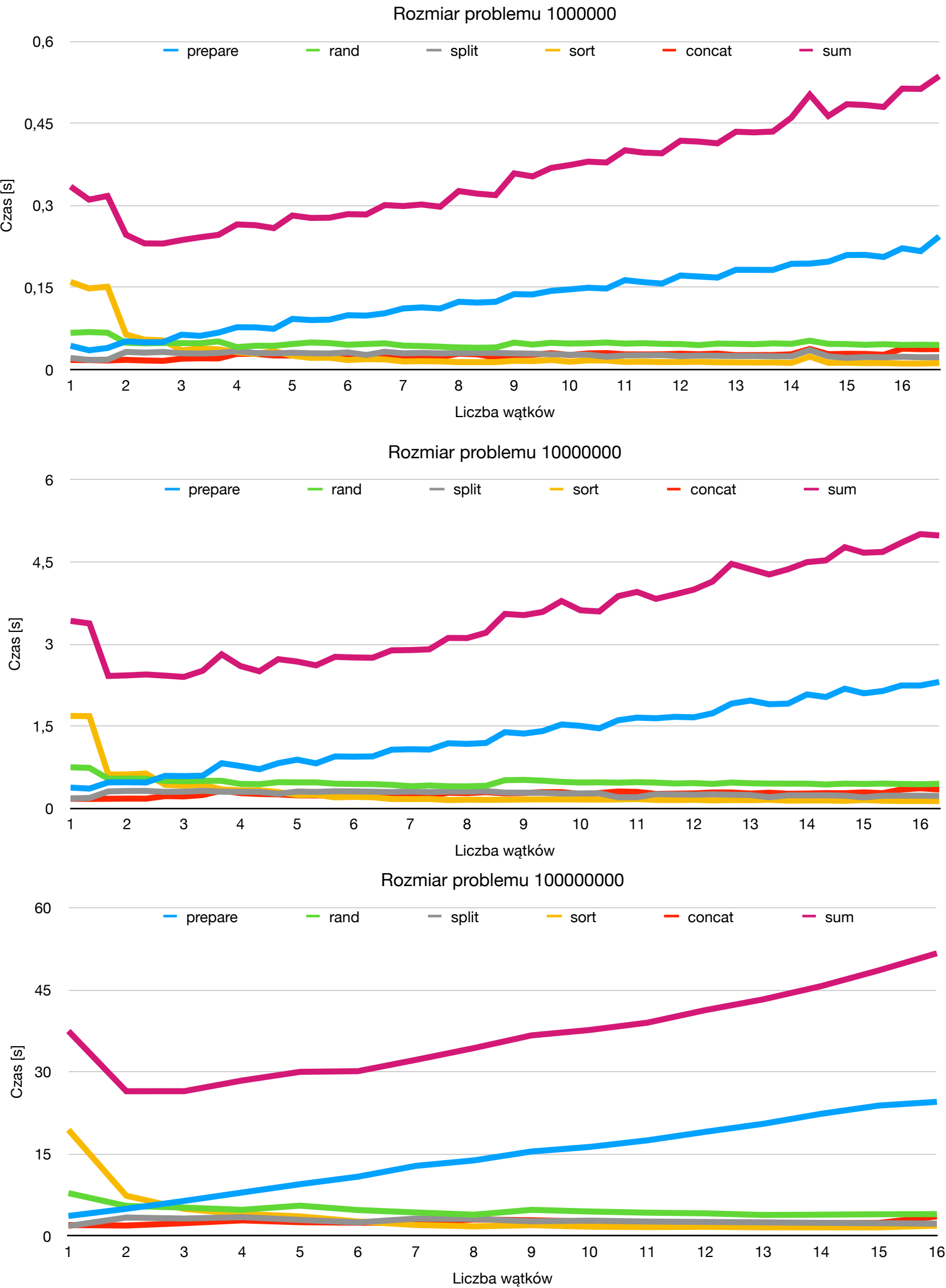


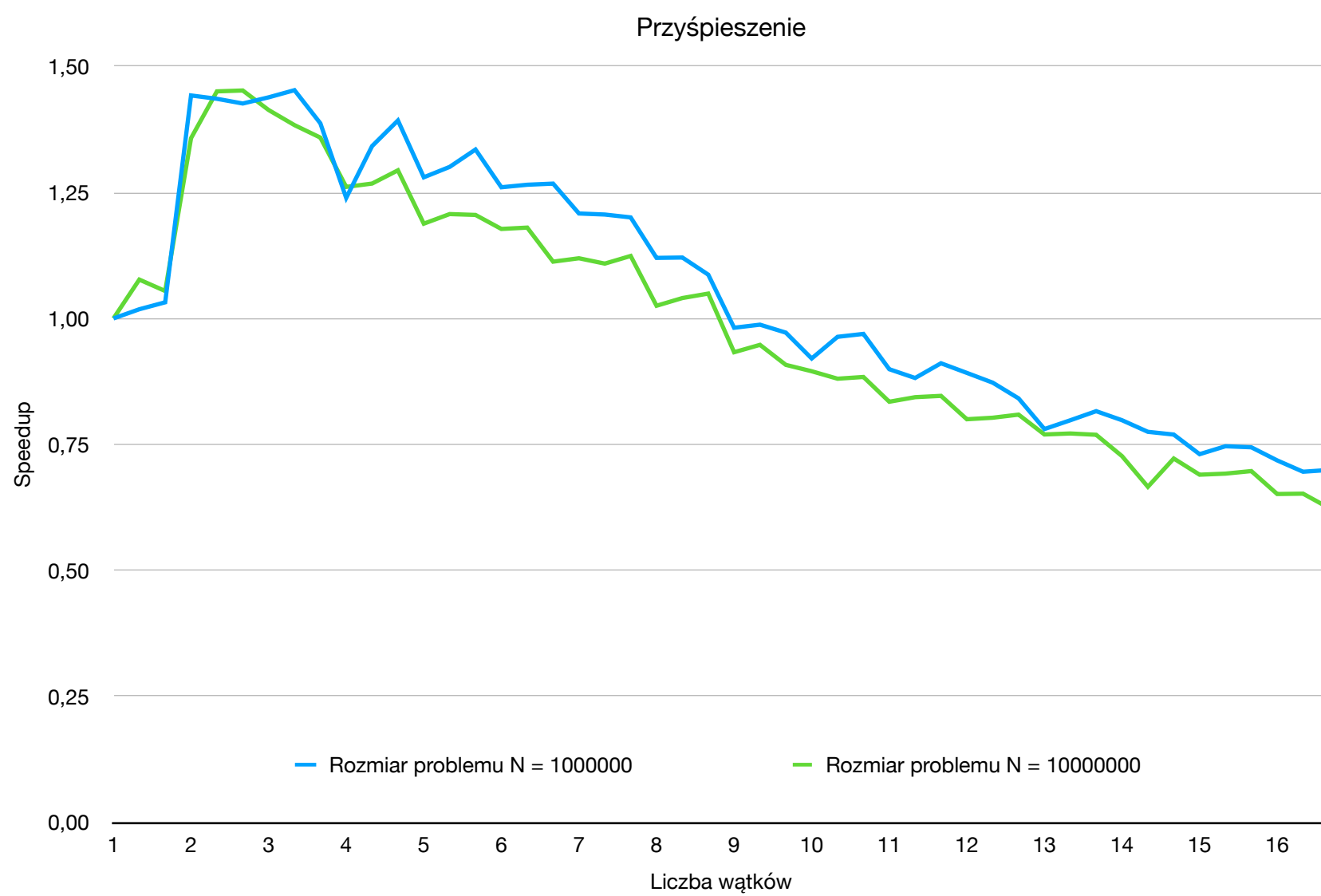
Algorytm sekwencyjny 2

### Test przyspieszenia

Zgodnie w testami można zaobserwować, że wcześniej omawiane zagadnienia związane z efektywnością w *Pytanie 2* są prawdziwe. Niezależnie od rozmiaru problemu, najlepsze przyspieszenie osiąga się dla 2-3 kubelków. Widać wyraźnie na wykresach, że czas

sortowanie maleje najmocniej na samym początku - wynika to z tego, że czas sortowania jest dzielony przez  $n$ , a dla większych  $n$  różnica jest mniejsza. Natomiast zgodnie z oczekiwaniem, czas przygotowania kubeków pod kątem wypełnienia ich wartościami rośnie liniowo względem liczby kubeków. Początkowo największy narzut stanowi sortowanie, natomiast wraz ze wzrostem liczby wątków sortowanie staje się najmniej czasochłonnym etapem, a przygotowanie kubeków stanowi połowę całego czasu wykonywania algorytmu. Widzimy zatem, że przyspieszenie nie jest wybitne i w najlepszym wypadku wynosi 1,45 dla dwóch procesorów (dla porównania dla 16 wątków przyspieszenie wynosi 0,7). Zdaje się, że graniczną liczbą wątków jest 3. Na wykresie przyspieszenia widać, że po przekroczeniu 3 wątków / 3 kubeków następuje stały spadek przyspieszenia.





# Algorytm 2 sekwencyjny - rozmiar problemu względem liczby kubełków

size	buckets	prepare	rand	split	sort	concat	sum
200000	1	0,008944988251	0,01677703857	0,003882169724	0,03072285652	0,003404855728	0,0686750412
200000	1	0,00767493248	0,01447606087	0,003846883774	0,02906107903	0,00345492363	0,06346011162
200000	1	0,006707906723	0,01747012138	0,004580974579	0,03352618217	0,004081964493	0,07235789299
200000	2	0,01079916954	0,01652312279	0,006770849228	0,02822399139	0,004573106766	0,07347202301
200000	2	0,01010704041	0,01460909843	0,006632804871	0,02868390083	0,004472970963	0,07089209557
200000	2	0,009336948395	0,01533699036	0,007075071335	0,03592395782	0,005348920822	0,08033490181
200000	3	0,01381707191	0,01677107811	0,00985789299	0,02895689011	0,005671024323	0,0839381218
200000	3	0,01225805283	0,01489496231	0,008955001831	0,03012681007	0,005752086639	0,08037304878
200000	3	0,01297211647	0,01474308968	0,008705854416	0,02714109421	0,005875110626	0,07765197754
200000	4	0,01466393471	0,01436591148	0,01019692421	0,02698802948	0,004964113235	0,0823738575
200000	4	0,01493406296	0,01542592049	0,01013612747	0,0267560482	0,004941940308	0,08239197731
200000	4	0,01578879356	0,01523280144	0,01006793976	0,02651000023	0,004910945892	0,08232307434
400000	1	0,01469516754	0,02980899811	0,007254123688	0,06231093407	0,006807088852	0,1307249069
400000	1	0,01482510567	0,02921009064	0,007730960846	0,05986690521	0,00676202774	0,1281399727
400000	1	0,01396298409	0,0290119648	0,007781982422	0,06220698357	0,006913900375	0,1301450729
400000	2	0,01929998398	0,03048300743	0,01384401321	0,05790281296	0,008929014206	0,143912077
400000	2	0,01888394356	0,02948784828	0,01354980469	0,05710482597	0,009041070938	0,1411418915
400000	2	0,01929593086	0,0300090313	0,01328206062	0,0576570034	0,008774995804	0,1424458027
400000	3	0,02547502518	0,03210091591	0,01773095131	0,05701184273	0,01156687737	0,1615509987
400000	3	0,02483296394	0,03396701813	0,02012300491	0,05910992622	0,01137900352	0,1660380363
400000	3	0,02284598351	0,03064012527	0,01676988602	0,05498981476	0,01130390167	0,1537380219
400000	4	0,03127193451	0,03241705894	0,02134203911	0,05885314941	0,01092600822	0,1763319969
400000	4	0,03188896179	0,03380489349	0,02116799355	0,05652618408	0,010201931	0,1734569073
400000	4	0,02814984322	0,02975702286	0,0204231739	0,05837392807	0,01092410088	0,1683111191
600000	1	0,02659010887	0,05482792854	0,01402211189	0,1093149185	0,01234316826	0,2348051071
600000	1	0,02437210083	0,05368089676	0,01303195953	0,1084709167	0,01231789589	0,2294340134
600000	1	0,02432513237	0,05361890793	0,01201796532	0,1065940857	0,0122718811	0,2263071537
600000	2	0,03221821785	0,05425310135	0,02401304245	0,1042640209	0,01565003395	0,2533740997
600000	2	0,03392696381	0,05384516716	0,02397704124	0,1034021378	0,0155749321	0,2536799908
600000	2	0,03251004219	0,05366802216	0,02334403992	0,09582304955	0,01407885551	0,2403841019
600000	3	0,04011297226	0,0503180027	0,02577710152	0,08378195763	0,01512694359	0,2439858913
600000	3	0,03927993774	0,04487490654	0,02526497841	0,08627605438	0,01895809174	0,2391331196
600000	3	0,03891611099	0,0454211235	0,02608704567	0,08555102348	0,01680016518	0,2381708622
600000	4	0,04741001129	0,04965686798	0,03107404709	0,08590698242	0,01539683342	0,2602939606
600000	4	0,04407286644	0,04812002182	0,03153991699	0,09842300415	0,01552319527	0,2711760998
600000	4	0,04861211777	0,04954409599	0,03203105927	0,08622097969	0,01681089401	0,2641649246
800000	1	0,03131198883	0,05811595917	0,01483392715	0,1327929497	0,01426410675	0,2719299793
800000	1	0,02885603905	0,06072402	0,01747298241	0,1309030056	0,01495790482	0,2734501362
800000	1	0,02999091148	0,05936288834	0,01534509659	0,1345951557	0,01413202286	0,2737598419
800000	2	0,04168915749	0,05864214897	0,02720499039	0,1294240952	0,01828885078	0,3025870323
800000	2	0,03959798813	0,0639629364	0,02806305885	0,1213519573	0,01738595963	0,2964229584
800000	2	0,03862905502	0,06022000313	0,02803111076	0,1280779839	0,02126002312	0,3042778969
800000	3	0,05502009392	0,06093883514	0,03616595268	0,1204659939	0,02299594879	0,3307049274
800000	3	0,04771399498	0,06142711639	0,03505086899	0,1174120903	0,0226111412	0,3169682026
800000	3	0,05398702621	0,06330513954	0,0352268219	0,1373169422	0,02390599251	0,3499200344
800000	4	0,06329894066	0,05854821205	0,04069590569	0,114566803	0,02013516426	0,3374938965
800000	4	0,06169009209	0,05788803101	0,03978800774	0,1104879379	0,02003908157	0,3298721313
800000	4	0,0635330677	0,06690883636	0,04718780518	0,1241509914	0,0234079361	0,3729841709

# Algorytm 2 równoległy - Test przyspieszenia

threads	prepare	rand	split	sort	concat	sum	speedup
1	0,04348707199	0,06715416908	0,02111101151	0,1601879597	0,01755595207	0,3346130848	1
1	0,0352268219	0,0685839653	0,01768302917	0,1483240128	0,01731300354	0,3106651306	1,07708607063093
1	0,03944993019	0,06732487679	0,01809096336	0,1514008045	0,01661705971	0,3173341751	1,05445020125726
2	0,05158209801	0,04887795448	0,03224110603	0,06408405304	0,01773881912	0,2464299202	1,35784276734104
2	0,04978919029	0,04794001579	0,0309510231	0,05470705032	0,01663589478	0,2306501865	1,45073840987334
2	0,05001020432	0,04858708382	0,03216814995	0,0533850193	0,01591300964	0,2303719521	1,45249055603241
3	0,06371808052	0,04835391045	0,02971315384	0,03515195847	0,01982998848	0,2366888523	1,4137255791662
3	0,061606884	0,04812502861	0,02929496765	0,03839802742	0,02061486244	0,2418169975	1,38374509757115
3	0,0675008297	0,05148291588	0,03029894829	0,03689694405	0,02031803131	0,2462229729	1,35898401704336
4	0,07729101181	0,04105210304	0,03345704079	0,03183913231	0,02878022194	0,2653329372	1,26110647374238
4	0,07716584206	0,04352092743	0,03030800819	0,02847790718	0,0287539959	0,2639510632	1,26770879701461
4	0,07456088066	0,04338908195	0,02957987785	0,03370594978	0,02579307556	0,2585878372	1,29400163759906
5	0,09262704849	0,04671502113	0,03074812889	0,02529191971	0,02568101883	0,2816359997	1,18810480604905
5	0,09049701691	0,04954218864	0,02973389626	0,02141499519	0,02408790588	0,2771959305	1,20713563217336
5	0,09117412567	0,04842805862	0,02947211266	0,02162694931	0,02228689194	0,2775871754	1,20543423635413
6	0,09928417206	0,04535889626	0,03133797646	0,01739907265	0,02130103111	0,2841470242	1,17760545176246
6	0,09867310524	0,04672217369	0,02661705017	0,01909089088	0,02223587036	0,2834980488	1,18030119154739
6	0,1028130054	0,04790997505	0,03253602982	0,0185251236	0,02448105812	0,3007171154	1,11271712737372
7	0,1117830276	0,04381203651	0,02963781357	0,01487112045	0,02194094658	0,2989161015	1,11942141330249
7	0,1136300564	0,04294514656	0,0303478241	0,01531386375	0,02169799805	0,3017799854	1,10879813436428
7	0,1115880013	0,04179501534	0,02958106995	0,01495599747	0,0230050087	0,2977011204	1,12399000833589
8	0,1239440441	0,0404419899	0,03015804291	0,01355290413	0,02894496918	0,3263990879	1,025165502002
8	0,1224148273	0,03988790512	0,03040599823	0,01359200478	0,02786207199	0,3216021061	1,04045675837693
8	0,1238560677	0,04034304619	0,03032803535	0,01368713379	0,02202296257	0,3188910484	1,04930221929679
9	0,1379339695	0,0492811203	0,02954006195	0,01600503922	0,02686095238	0,3586819172	0,932896443211049
9	0,1373119354	0,04568696022	0,02907395363	0,01549196243	0,02707910538	0,3531279564	0,947568944161907
9	0,1438231468	0,04883599281	0,02795481682	0,01753902435	0,02992987633	0,3685929775	0,907811882552754
10	0,1465759277	0,04758810997	0,02714896202	0,01436495781	0,02655887604	0,3738100529	0,895142017193193
10	0,1495571136	0,04812002182	0,02708601952	0,01676797867	0,0295188427	0,3801100254	0,880305865250141
10	0,1481630802	0,04938101768	0,02356290817	0,01652598381	0,03012299538	0,3786180019	0,883774894803807
11	0,1634991169	0,04742789268	0,02579307556	0,01387906075	0,0277659893	0,4009230137	0,83460682815874
11	0,1597909927	0,04824399948	0,02614998817	0,01437902451	0,02763390541	0,3966221809	0,843657013938829
11	0,1571910381	0,04699397087	0,0262761116	0,01372504234	0,02784395218	0,3953990936	0,846266696651831
12	0,1719329357	0,04666495323	0,02469491959	0,01381492615	0,02894020081	0,4183452129	0,799849202242419
12	0,1700799465	0,04472613335	0,02552390099	0,01415586472	0,02793908119	0,4167728424	0,802866815585007
12	0,1681950092	0,04746603966	0,0250351429	0,0132791996	0,02919793129	0,4135890007	0,809047349503171
13	0,1823680401	0,04697608948	0,02479195595	0,01309204102	0,02625608444	0,4347810745	0,769612810734245
13	0,1822810173	0,04625701904	0,02477788925	0,01283407211	0,02650594711	0,4336650372	0,771593409882571
13	0,1821448803	0,04786801338	0,02434301376	0,01296806335	0,02650094032	0,4351260662	0,769002619682636
14	0,1933128834	0,0470559597	0,02419686317	0,01218509674	0,02790617943	0,4604980946	0,726632940991109
14	0,1937580109	0,05273795128	0,03566789627	0,02368187904	0,03761506081	0,5026190281	0,665738991348784
14	0,1972239017	0,04700088501	0,02409100533	0,01230192184	0,02813696861	0,4635000229	0,721926792379453
15	0,209373951	0,04655385017	0,02135396004	0,01241803169	0,02882695198	0,4851150513	0,689760261825131
15	0,209624052	0,04525184631	0,02292609215	0,01171112061	0,02853512764	0,4836239815	0,691886874100349
15	0,2058272362	0,04632687569	0,02248215675	0,01186704636	0,0270922184	0,4801051617	0,696957899005234
16	0,2217590809	0,04507899284	0,02336001396	0,01103591919	0,03807997704	0,513646841	0,651445814693524
16	0,216531992	0,04535007477	0,02251505852	0,01096916199	0,03721404076	0,5131230354	0,652110822775975
16	0,2433991432	0,0450720787	0,02266812325	0,01188516617	0,0374751091	0,5363829136	0,623832482944326

Rozmiar problemu N = 1000000

## Algorytm 2 równoległy - Test przyspieszenia

threads	prepare	rand	split	sort	concat	sum	speedup
1	0,3860881329	0,7678530216	0,1800730228	1,713555098	0,1786789894	3,484699011	1
1	0,3752989769	0,7467460632	0,1772010326	1,68522191	0,1713631153	3,422367096	1,01821310024657
1	0,3561520576	0,735200882	0,1852040291	1,680231094	0,1682901382	3,376703024	1,03198267251589
2	0,4706771374	0,5346908569	0,3049480915	0,6126458645	0,1704699993	2,415507078	1,44263663838455
2	0,4725601673	0,5399129391	0,3128020763	0,6159241199	0,1740100384	2,426681042	1,43599383301236
2	0,4681580067	0,5357990265	0,3141419888	0,6307570934	0,1725921631	2,442584038	1,42664447027718
3	0,5889840126	0,496450901	0,2965769768	0,4251129627	0,2186379433	2,421793938	1,4388916234045
3	0,5847499371	0,4925551414	0,3012428284	0,4118599892	0,2134931087	2,39810586	1,45310474784462
3	0,5921959877	0,4987499714	0,3158359528	0,4279341698	0,2390828133	2,511771917	1,38734691132388
4	0,8206100464	0,4984531403	0,2997770309	0,3412339687	0,3200740814	2,811941147	1,2392503359175
4	0,7684988976	0,4414770603	0,2972719669	0,3250260353	0,2741999626	2,596586943	1,34203055298965
4	0,7108500004	0,4365320206	0,2948899269	0,3234651089	0,2584388256	2,501646996	1,39296192331366
5	0,821171999	0,4734668732	0,2613561153	0,2942278385	0,2516639233	2,722080946	1,28015995120227
5	0,8855650425	0,470334053	0,3025529385	0,2390220165	0,2326509953	2,678913116	1,30078836457494
5	0,8179030418	0,470621109	0,2963578701	0,241300106	0,2312378883	2,609692097	1,33529124566299
6	0,9442858696	0,4461259842	0,3079040051	0,1969249249	0,229598999	2,764834881	1,26036423908962
6	0,9405930042	0,4414730072	0,3053710461	0,2053778172	0,2198650837	2,753895998	1,26537059261887
6	0,9447300434	0,4379041195	0,2995238304	0,198141098	0,2320768833	2,748983145	1,26763200325115
7	1,067199945	0,4250371456	0,2933449745	0,1669678688	0,2187800407	2,883951902	1,20830690989797
7	1,076163054	0,4000909328	0,3031151295	0,1650550365	0,2176241875	2,888909101	1,20623352593329
7	1,070219994	0,41284585	0,2913200855	0,1667878628	0,2182760239	2,902608156	1,20054062543604
8	1,18534112	0,399228096	0,3021638393	0,148045063	0,2813570499	3,110814095	1,12018876878594
8	1,173731804	0,3975980282	0,3012318611	0,1533041	0,2733268738	3,108889818	1,12088212030678
8	1,190975189	0,407725811	0,3100891113	0,15107584	0,2927489281	3,206171989	1,08687214003353
9	1,387898207	0,5100548267	0,2830529213	0,1524250507	0,2692689896	3,550391197	0,981497197814284
9	1,362713099	0,5159060955	0,282022953	0,1583819389	0,2755889893	3,528470993	0,987594631757824
9	1,408045053	0,5017430782	0,278575182	0,1589918137	0,2916910648	3,58575201	0,971818185218001
10	1,528894901	0,4792339802	0,2655260563	0,1590309143	0,2927398682	3,785860062	0,920451087449624
10	1,503574133	0,4679429531	0,2683308125	0,1577031612	0,2557418346	3,616774082	0,963482631758143
10	1,458678007	0,4708929062	0,2713170052	0,1568448544	0,2759990692	3,595227003	0,969257019957913
11	1,60622406	0,4648590088	0,1976079941	0,1579501629	0,3033430576	3,875445127	0,899173874691788
11	1,654067993	0,4737651348	0,1985177994	0,1650202274	0,2965378761	3,952152014	0,881721907116906
11	1,642534971	0,4666109085	0,2555031776	0,1497659683	0,2594969273	3,82517004	0,910991923119841
12	1,668550968	0,4497320652	0,2495129108	0,14757514	0,2631242275	3,907191992	0,891867872921255
12	1,659715176	0,4574780464	0,2442538738	0,1522159576	0,268089056	3,994023085	0,872478435111499
12	1,733651161	0,4420580864	0,2513659	0,1414179802	0,2859230042	4,142260075	0,841255485630028
13	1,91027689	0,4667720795	0,2437348366	0,1473369598	0,2857618332	4,465398073	0,780378132930681
13	1,966319084	0,4541480541	0,2384231091	0,1453280449	0,2652711868	4,367746115	0,797825450300927
13	1,899188995	0,4482729435	0,1980259418	0,15260005	0,2817101479	4,271010876	0,815895606958412
14	1,90988493	0,4465119839	0,233987093	0,1361489296	0,2615859509	4,367233992	0,797919007175561
14	2,078202009	0,4469411373	0,2335641384	0,1400010586	0,2657129765	4,49699688	0,774894691721467
14	2,03094697	0,4318211079	0,2342550755	0,1398458481	0,2742300034	4,528789997	0,769454757961479
15	2,183548927	0,4471921921	0,2265059948	0,1337230206	0,2721748352	4,770313978	0,730496782197342
15	2,098755121	0,4413878918	0,1954760551	0,1478128433	0,2873489857	4,669531107	0,746263153869166
15	2,140047073	0,448112011	0,2261371613	0,1351919174	0,2723460197	4,682723999	0,744160666258392
16	2,241969109	0,4406790733	0,2270038128	0,1299009323	0,3405089378	4,852308989	0,718152743137273
16	2,241168022	0,4363799095	0,2271409035	0,1296989918	0,3622469902	5,00931406	0,695643948305369
16	2,306197882	0,4426839352	0,2217140198	0,1283090115	0,3393318653	4,984000921	0,699177039939395

Rozmiar problemu N = 10000000



### Algorytm 2 równoległy - Testy przyspieszenia

threads	prepare	rand	split	sort	concat	sum
1	3,648807049	7,8257339	1,813508034	19,39783192	1,988012075	37,46217918
2	4,956936836	5,481914043	3,357952118	7,342695951	1,898709059	26,45170116
3	6,39724493	5,18153882	3,178894043	4,937428951	2,333494902	26,46180201
4	7,934552908	4,775122881	3,431035995	4,001235962	2,840535879	28,40834498
5	9,456007004	5,532356024	2,923340082	3,571358919	2,508902073	29,99321914
6	10,82885695	4,731510162	2,475205898	2,634702921	2,414669037	30,11871696
7	12,80553913	4,291516066	3,162973881	1,999948025	2,376348019	32,18883109
8	13,79924107	3,882989883	3,094436884	1,751938105	3,009535074	34,33365393
9	15,43814898	4,765244961	2,673628092	1,96177721	2,865499973	36,66785097
10	16,26664996	4,468197107	2,796416998	1,685675144	2,629936934	37,64687204
11	17,47326398	4,251375914	2,625880957	1,613764048	2,54229188	38,98571706
12	19,03179598	4,127944231	2,553526163	1,664716959	2,455705881	41,27734208
13	20,49277115	3,830820084	2,468394041	1,578612089	2,432461977	43,24600816
14	22,32637501	3,883532047	2,389365196	1,566720009	2,313101053	45,6644001
15	23,83597612	3,958362103	2,319844007	1,56927681	2,376232862	48,55233908
16	24,51209188	3,983898163	2,246116161	1,884553909	3,558399916	51,66992307

Rozmiar problemu N = 10000000