

# Sprawozdanie Lab2

## Powiększanie tablicy dwoma sposobami.

Pliki do kompilacji oraz plik Makefile w folderze o nazwie Lab2.

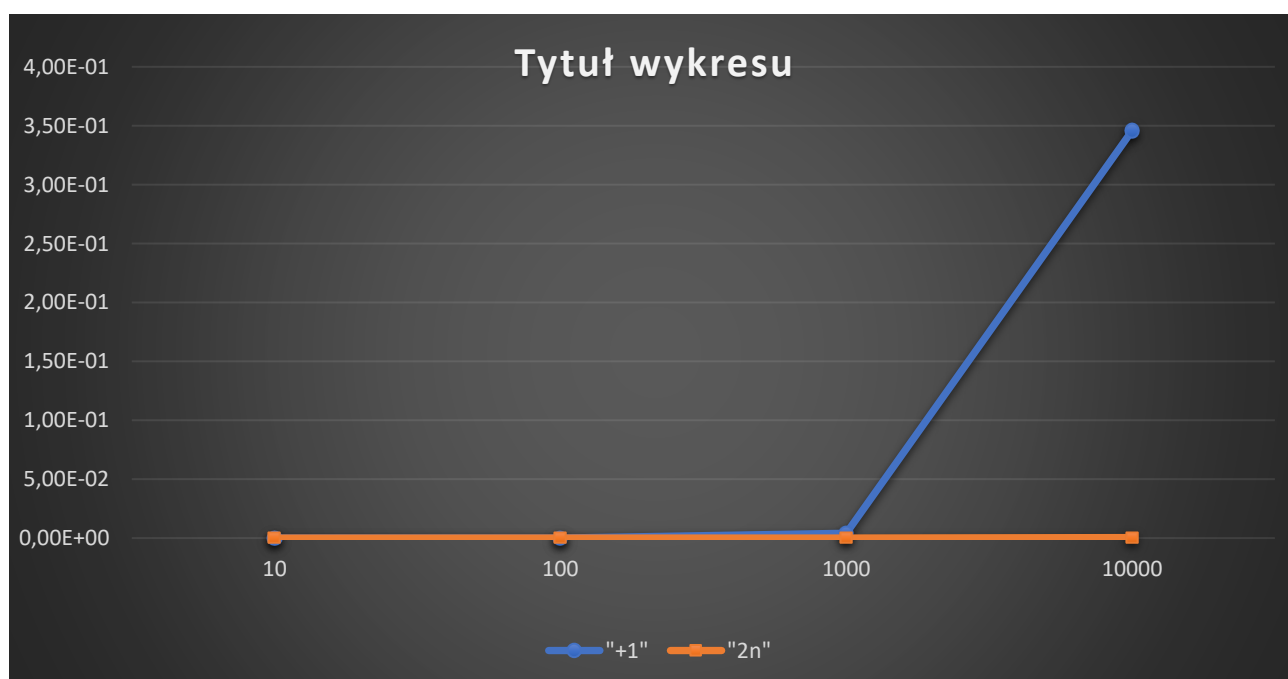
Niżej tabele z pomiarami czasów wykonywania algorytmu dla powiększania tablicy 10-elementowej o:  $10^1, 10^2, 10^3, 10^4$  elementów. Tylko na tyle pozwoliła specyfikacja komputera (AMD A4-4300M APU 2.50 GHz).

		Czas algorytmu "2n" [s]			
		10	100	1000	10000
	1	0,0000080	0,0000110	0,0000920	0,0003990
	2	0,0000060	0,0000100	0,0000450	0,0003620
	3	0,0000080	0,0000100	0,0000580	0,0003540
	4	0,0000070	0,0000140	0,0000710	0,0003320
	5	0,0000080	0,0000170	0,0000600	0,0003980
	6	0,0000090	0,0000150	0,0000520	0,0006160
	7	0,0000100	0,0000170	0,0000900	0,0004180
	8	0,0000080	0,0000090	0,0000730	0,0006670
	9	0,0000070	0,0000120	0,0000530	0,0005280
	10	0,0000080	0,0000110	0,0000580	0,0004410
	11	0,0000080	0,0000120	0,0000650	0,0005800
	12	0,0000080	0,0000170	0,0000490	0,0006200
	13	0,0000080	0,0000170	0,0000960	0,0003840
	14	0,0000080	0,0000130	0,0000470	0,0005330
	15	0,0000080	0,0000110	0,0000430	0,0003480
	16	0,0000050	0,0000150	0,0000440	0,0004240
	17	0,0000120	0,0000090	0,0000440	0,0005200
	18	0,0000070	0,0000080	0,0000660	0,0003950
	19	0,0000110	0,0000100	0,0000630	0,0003410
	20	0,0000050	0,0000100	0,0000440	0,0003710

		Czas algorytmu "+1" [s]			
		10	100	1000	10000
1		0,000006	0,000057	0,003485	0,345954
2		0,000015	0,000067	0,003957	0,347412
3		0,000010	0,000120	0,003636	0,347438
4		0,000015	0,000100	0,003608	0,340633
5		0,000006	0,000084	0,004041	0,338552
6		0,000013	0,000120	0,003610	0,343252
7		0,000007	0,000100	0,003605	0,344621
8		0,000059	0,000100	0,004739	0,350970
9		0,000014	0,000110	0,003566	0,345906
10		0,000006	0,000089	0,005248	0,344370
11		0,000010	0,000077	0,005274	0,347747
12		0,000012	0,000084	0,005543	0,345437
13		0,000008	0,000091	0,004096	0,337838
14		0,000014	0,000084	0,004554	0,346841
15		0,000016	0,000100	0,003917	0,350552
16		0,000011	0,000110	0,003605	0,345347
17		0,000017	0,000084	0,003582	0,345973
18		0,000009	0,000067	0,004632	0,350352
19		0,000010	0,000061	0,005662	0,351398
20		0,000007	0,000091	0,003869	0,342302

Czasy średnie przeprowadzonych testów:

Zwiększano o:	$10^1$	$10^2$	$10^3$	$10^4$
$T_{sr}(2n)$ [s]	0,00000795	0,0000124	0,00006065	0,00045155
$T_{sr}(+1)$ [s]	0,00001325	0,0000898	0,00421145	0,34564475



## Wnioski:

- Widoczne są nieznaczne różnice w czasie na początku, w szczególności w tabeli samych pomiarów. Algorytm zwiększania sukcesywnego „+1” jest szybszy dla małych powiększeń (dla 10 i 100 lepszy jest algorytm „+1”)
- Przy bardzo dużym powiększaniu rozmiaru tablicy zdecydowanie lepszy okazał się algorytm „2n”. Jednak nie jest on idealny ponieważ potrzeba dla niego więcej pamięci, więc jest szybszy, ale mniej ekonomiczny.
- Dla małych rozmiarów algorytm „2n” jest nieekonomiczny, a dla dużych o wiele szybszy.
- Z kolei algorytm „+1” dla małych wielkości, bardzo ekonomiczny, lecz dla dużych bardzo czasochłonny.