

Laboratorium nr2.

1. Wstęp

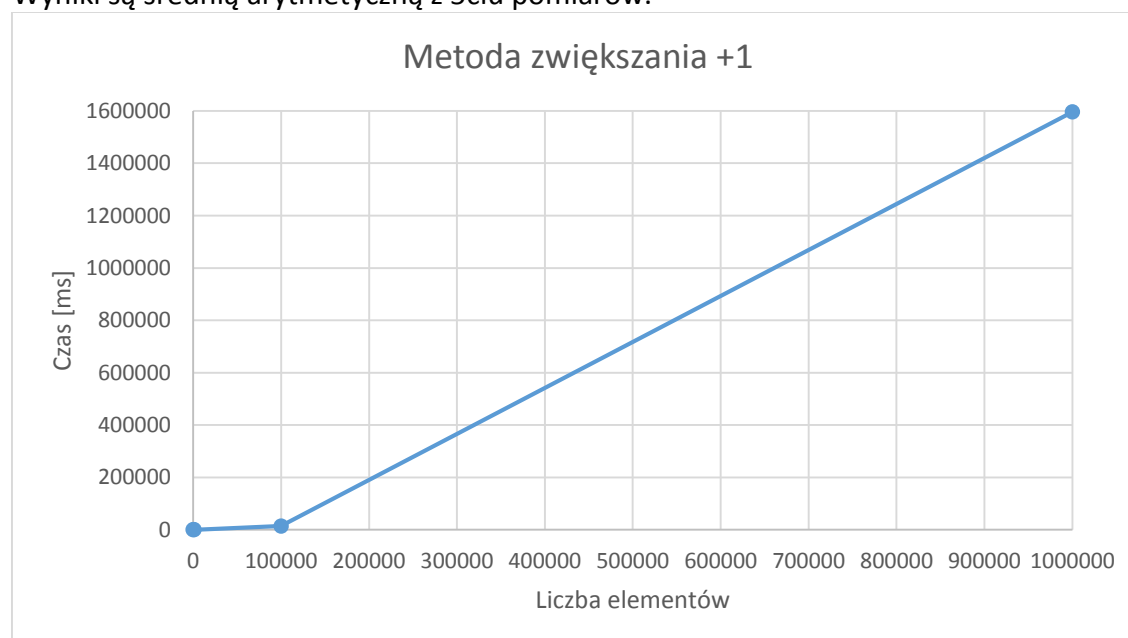
W tym ćwiczeniu mieliśmy za zadane zaprojektować tablice dynamiczną i metody zwiększania jej rozmiaru dodając jeden element lub mnożąc jej wcześniejszy rozmiar razy dwa w zależności od ilości elementów podanych przez użytkownika. Liczba elementów miała się mieścić w przedziale 10-1000000000. Kolejnym wymogiem było mierzenie czasu zwiększania rozmiaru tej tablicy i wyświetlenie go na ekranie końcowym. Czas podawany miał być w milisekundach.

2. Wyniki i wykresy

Metoda zwiększania rozmiaru tablicy $n+1$, gdzie n to początkowy rozmiar alokowanej tablicy.

Ilość elementów	Czas
10	0[ms]
1000	1[ms]
100000	14321[ms]
1000000	1595890[ms]
1000000000	-----

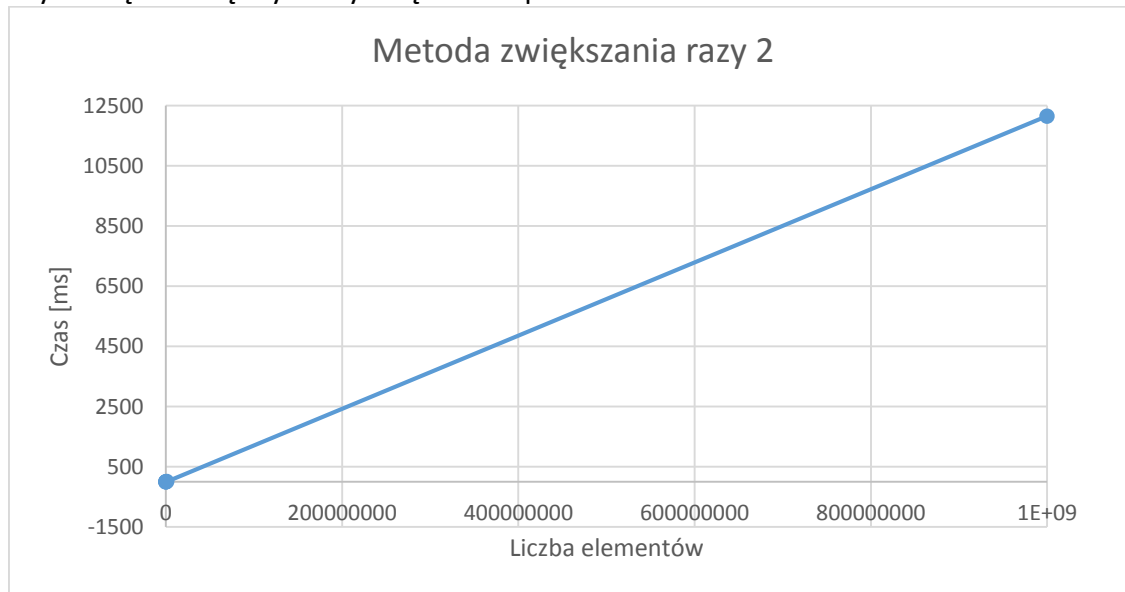
Wyniki są średnią arytmetyczną z 5ciu pomiarów.



Metoda zwiększania rozmiaru tablicy $n*2$.

Ilość elementów:	Czas:
10	0[ms]
10	0[ms]
1000	0[ms]
100000	1[ms]
1000000	12[ms]
1000000000	12154[ms]

Wyniki są średnią arytmetyczną z 10ciu pomiarów.



3. Wnioski

Wyniki pomiarowe, którymi się posłużyłem w zadaniu zrobione są na bazie kolejnego zadania(więcej o tym w Informacji o problemie).

Biorąc pod uwagę powyższe wyniki i wykresy , możemy zauważyć, że metoda zwiększania rozmiaru tablicy dynamicznej „razy dwa” jest o wiele bardziej efektywna , niż „zwiększanie +1” rozmiaru naszej tablicy. Najlepiej różnice widać na pomiarach wykonanych przy dużej ilości elementów tj 1mln , czy 1mld. Dla takich wartości pierwszy sposób miał spore problemy, a w przypadku miliarda nie dał rady wykonać zadania przy wymaganiach sprzętowych komputera na którym pracuje. Zaś algorytm zwiększania rozmiaru tablicy razy dwa poradził sobie bardzo dobrze nawet przy 1mld elementów. Początkowo miałem wątpliwości co do wyników czasowych jakie otrzymałem, lecz po uświadomieniu sobie rozmiaru alokowanej pamięci(jeden element typu int zajmuje 4bity pamięci) uzyskane wyniki okazały się być bardzo prawdopodobnymi wartościami o czym dodatkowo świadczy liniowy wzrost czasu do liczby elementów na powyższych wykresach.

Informacja o problemie.

Po zmianie metody mierzenia czasu na `gettimeofday`, program przy próbie kompilacji wyrzucił mi `Segmentation Fault`. Początkowo wydawało mi się, że źle mierze czas lub nie czyszcze miejsca w pamięci po wykonaniu tablicy. Zmieniłem więc funkcje mierzenia czasu na tę z biblioteki `chrono` oraz stworzyłem funkcję `clean`(dodatkowo), która ma usuwać tablice po wyświetleniu wyników. Problem dalej występował. Po przeliczeniu konkretnych wartości na kartce stwierdziłem, że nie wychodzę poza rozmiar tablicy, a pomimo tego błąd `Segmentation Fault` dalej występuje. Po długim czasie googlowania i zmieniania wszystkiego co możliwe błąd nie ustępuje. Najciekawsze jest to, że po skorzystaniu z funkcji zwiększania tablicy jaka napisałem do kolejnego zadania (wiedziałem, że ta nie jest poprawna i mało efektywna, więc napisałem nową funkcję łatwiejszą do modyfikacji dla kolejnego zadania) również występuje w.w błąd. Wykorzystałem wyniki z kolejnego zadania, aby poprawić moje sprawozdanie według uwag od Pana uzyskanych.