

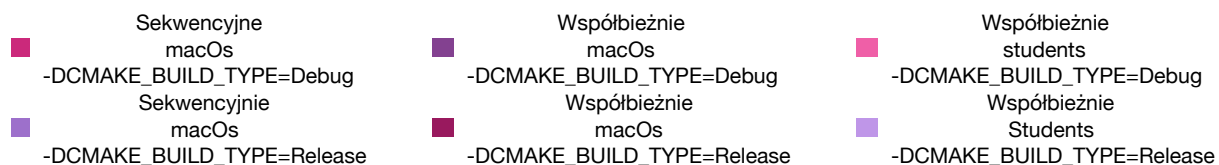
Analiza czasu działania algorytmów

0. Środowiska użyte do analizy:

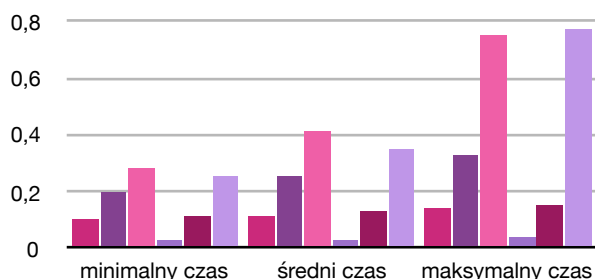
- MacBook Pro (15-inch, 2016)
macOS Catalina 10.15.3
2,6 GHz Quad-Core Intel Core i7
16 GB 2133 MHz LPDDR3
- Maszyna Students

Każdy test składał się z **10** uruchomień programu na środowisku.

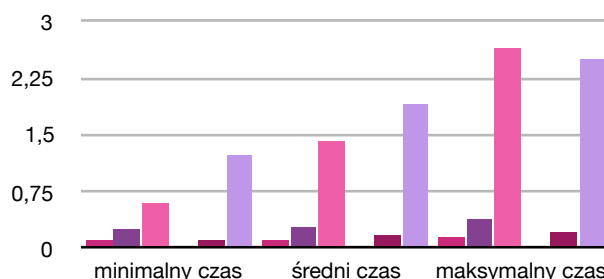
1. Znajdowanie maksymalnej wartości w tablicy



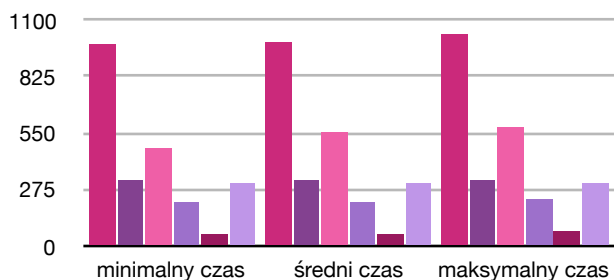
1.1 rozmiar tablicy = 100, liczba wątków = 4, losowe dane



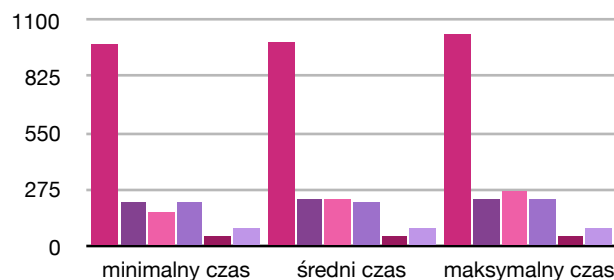
1.2 rozmiar tablicy = 100, liczba wątków = 16, losowe dane



1.3 rozmiar tablicy = 1 000 000, liczba wątków = 4, losowe dane



1.4 rozmiar tablicy = 1 000 000, liczba wątków = 16, losowe dane



WNIOSKI:

Współbieżne rozwiązania dla małych danych wejściowych wypadają gorzej niż sekwencyjne rozwiązania. Jest to spowodowane czasem potrzebnym na utworzenie i zarządzanie dodatkowymi wątkami i koniecznością walki o dostęp do pamięci. Dla dużych danych wejściowych widać znaczącą przewagę algorytmów współbieżnych, gdyż czas tworzenia i dostęp do pamięci stają się pomijalne. Widać wzrost wydajności wraz z liczbą wątków, jak i poprawę czasu rozwiązań kompilowanych z flagą *release* nad rozwiązaniami z flagą *debug*.