Krzysztof Kucharczyk
200401
Wydział Elektroniki
Kierunek AiR
PAMSI lab. czw. 10:00-13:15

Sprawozdanie z laboratorium nr 3
(Listy i stosy)

1 Opis zadania

Zadanie polegało na stworzeniu pięciu algorytmów, tj.

1. Stosu na bazie tablicy (powiększanie tablicy o jedno miejsce)
2. Stosu na bazie tablicy (podwajanie tablicy)
3. Stosu na bazie listy jednokierunkowej
4. Kolejki na bazie tablicy
5. Kolejki na bazie listy jednokierunkowej

oraz przetestowaniu wydajności dwóch pierwszych algorytmów.

1.1 Opis algorytmu 1

Algorytm ten miał działać jak zwykły stos. Problem przepełnienia stosu rozwiązany jest poprzez tworzenie nowej, większej tablicy, do której przepisywana jest stara tablica, a następnie usuwana. Nowa tablica zawiera jedno miejsce dodatkowo, w które wpisywany zostaje nowy element.

Usuwanie elementu z listy następuje poprzez stworzenie mniejszej tablicy, do której przepisywane są elementy starej poza ostatnim, następnie stara tablica zostaje usunięta i nadpisana nową.

1.2 Opis algorytmu 2

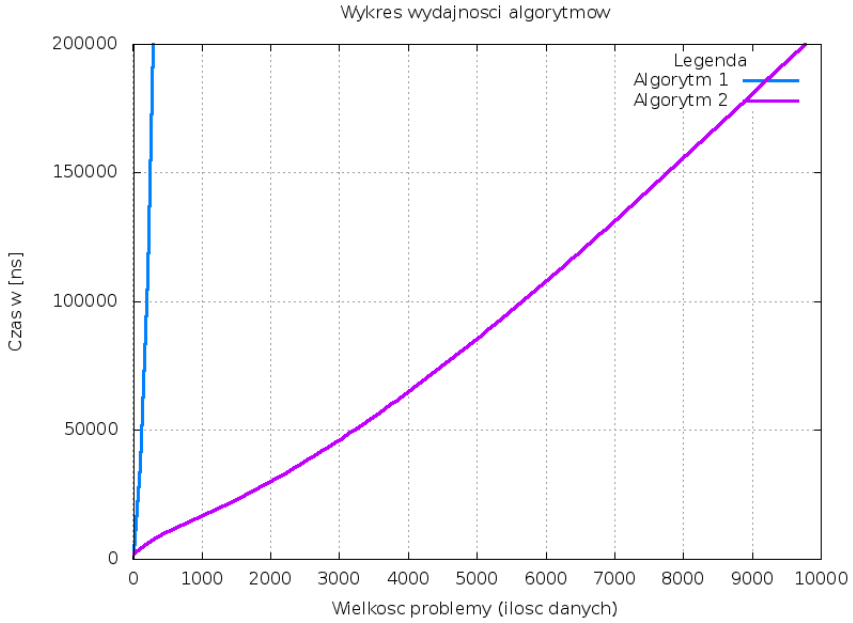
Algorytm zbudowany jest bardzo podobnie do poprzedniego, jednak tym razem tablica zostaje powiększona dwukrotnie w przypadku przepełnienia, natomiast zmniejszana o połowę w przypadku gdy ilość danych wynosi 1/4

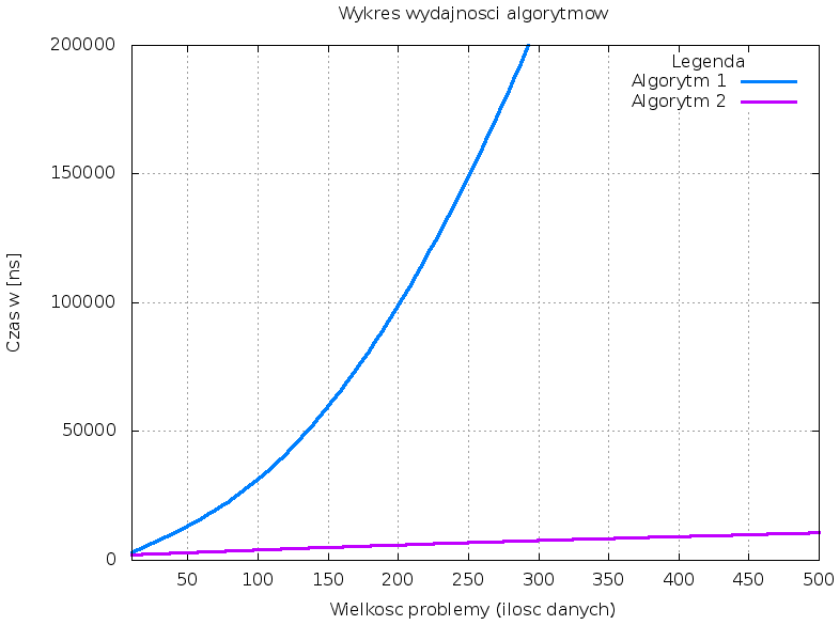
wielkości tablicy.

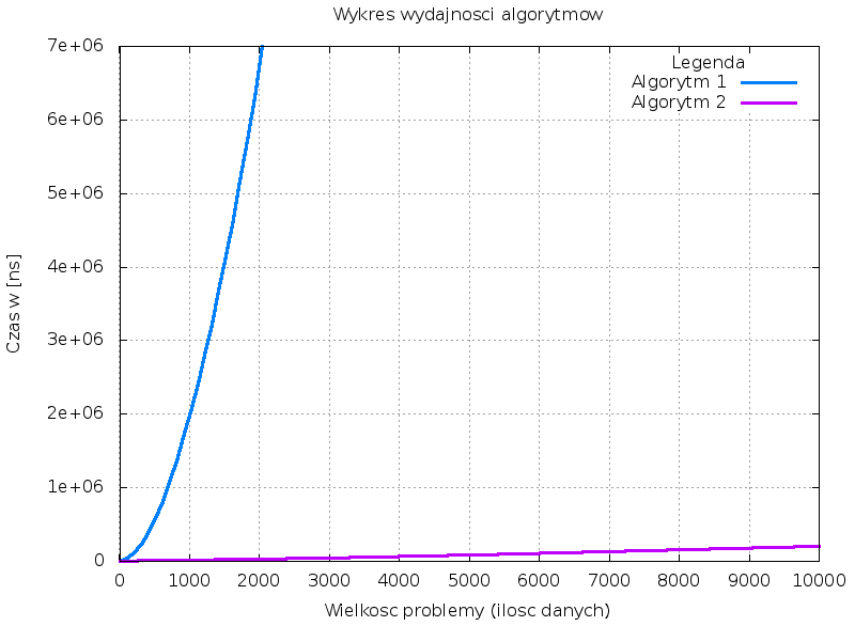
Oba algorytmy posiadają identyczny sposób mierzenia czasu, tj. pomiar odbywa się za pośrednictwem odpowiedniej biblioteki, która tworzy tymczasowy plik, w którym przechowywane są kolejne wyniki pomiarów. Program ./benchmark wyznacza średnią z owych pomiarów i umieszcza ją w pliku wynikowym.

2 Pomiary wydajności algorytmów

Na następnej stronie zamieszczone są wykresy wydajności algorytmów 1 i 2, zgodnie z przjętą powyżej konwencją. Wykres algorytmu 1 (zwiększający tablicę o jeden) ewidentnie szybciej rośnie w górę, przez co okazuje się być gorszym algorytmem. Z kolei algorytm 2 (zwiększający dwukrotnie wielkość tablicy) jest niemalże niewidoczny, gdyż znajduje się bardzo blisko osi rzędnych, czyli nie wymaga dużego czasu, aby uporać się ze sporą ilością danych. Dla prostszej analizy wstawiłem wykresy dla trzech zakresów (Umożliwia to obiektywne ocenie który z algorytmów jest wydajniejszy. Poprzez zaobserwowanie wydajności rozumiem zauważenie, który z algorytmów wolniej pnie się do góry wykresu, czyli potrzebuje mniej czasu na rozwiązanie danego problemu).







3 Wnioski

Wnioski z przeprowadzonej analizy są bardzo proste - algorytm podwajający wielkość tablicy przy jej przepełnieniu i zwalniający gdy jedna-czwarta jest zajęta jest o wiele bardziej wydajniejszym i efektywniejszym stosem. Wykres tego algorytmu idący niemalże przy osi z pewnością za tym przemawia. Drugi

algorytm również spełnia swoje zadanie, jednak prędko pnie się ku górze, co przemawia na jego niekorzyść - oznacza to o wiele dłuższy czas działania dla takiego samego zestawu danych jak dla algorytmu drugiego, co jest wyraźnie odczuwalne podczas testowania algorytmów.

Próbując odpowiedzieć na pytanie: "Dlaczego akurat drugi algorytm jest lepszy?"myślę, że zajmowanie nowego kawałka pamięci dla każdego nowego elementu i zwalnianie jej musi być bardzo obliczenio-chłonne, a przez to o wiele wolniejsze. Za to rzadkie, acz konkretne zwiększanie pamięci i zwalnianie jej jest wyjściem optymalnym, pozwalającym na ulepszenie działania mojego stosu.