

Projektowanie algorytmów i metody sztucznej inteligencji

Dawid Marszałkiewicz - 218665

6 marzec 2016

1 Zadanie I

- 1.1 Stwórz klasę pozwalającą na przechowywanie dowolnej ilości elementów w tablicy typu int
- 1.2 Wykonaj pomiary czasów zapisy do tablicy (obiek-
tu z 1.1) dla ilości elementów $n = 10^1, 10^3, 10^5, 10^6, 10^8$

2 Wyniki pomiaru

- 2.1 Wyniki dla strategii pierwszej polegającej na zwiększaniu tablicy o 1, gdy jest zapełniona

| lp. | Czas wykonania programu [s] |
|---------|--------------------------------|
| 10 | 0,0009 |
| 1000 | 0,0053 |
| 10000 | 0,4370 |
| 100000 | 45,6573 |
| 1000000 | 4807,3036 |

Uwaga: Zostały zmienione ilości elementów dla tej strategii, wyjaśnienie w pkt. 3

- 2.2 Wyniki dla strategii drugiej polegającej na zwiększaniu tablicy dwa razy, gdy jest zapełniona

| lp. | Czas wykonania programu [s] |
|-----------|--------------------------------|
| 10 | 0,0009 |
| 1000 | 0,0010 |
| 100000 | 0,0047 |
| 1000000 | 0,0394 |
| 100000000 | 4,7752 |

2.3 Wyniki dla strategii trzeciej polegającej na zwiększaniu tablicy dziesięć razy, gdy jest zapełniona

| lp. | Czas wykonania programu [s] |
|-----------|--------------------------------|
| 10 | 0,0013 |
| 1000 | 0,0014 |
| 100000 | 0,0049 |
| 1000000 | 0,0256 |
| 100000000 | 2,1699 |

3 Wnioski

- Po 14 godzinach ciągłej pracy programu w strategii I zostały zebrane wyniki dla miliona liczb, a następnie przybliżone wielomianem stopnia 2 i ekstrapolowane. Dla 100 milionów elementów czas działania programu wyszedłby po za okres siedmiu dni dzielący zajęcia laboratoryjne.
- Początkowo testy były przewidziane na miliard elementów, jednak zasoby pamięci komputera nie pozwoliły na ich poprawne przeprowadzenie, został zgłaszany wyjątek *bad_alloc*.
- Najszybsza była strategia polegająca na 10-krotnym zwiększaniu rozmiaru tablicy, a najwolniejsza na zwiększaniu tablicy o 1 element.