Звіт про експеримент

студента КП - 62

Благодатного Дмитра Романовича

Лабораторна робота: “Емпіричний аналіз алгоритмів”

**Частина 1.**

1. Опис середовища:

|  |  |  |
| --- | --- | --- |
|  | Комп’ютер №1 | Комп’ютер №2 |
| Апаратна частина | HP X302UJ | ACER aspire v3-571g |
| Операційна система | Linux Ubuntu 17.04 | Linux Ubuntu 17.04 |
| Інтегроване середовище розробки | IntelliJ IDEA 2016.1.3 | Eclsipse 3.8 |
| Мова програмування | Java | Java |

2. Псевдокод алгоритму та асимптотичний аналіз:

|  |
| --- |
| MergeSort (A[0...n-1], p, r)  Input: Масив A[0...n-1] неупорядкованих елементів  p і r – індекси (p < r)  Output: Масив A[0...n-1] упорядкований за зростанням  if p < r  q = (p + r)/2  Розділення  MergeSort(A, p, q) Підкорення  MergeSort (A, q + 1, r) Підкорення  Merge (A, p, q, r ) Комбінування  Початковий виклик: MergeSort (A, 0, A.lenght -1) |

Асимптотичний аналіз:

Cкладність по часу:

Найкраща - O(nlogn)

Середня - O(nlogn)

Найгірша - O(nlogn)

Cкладність по додатковій памяті:

Нативна - O(n)

Седжвік - O(logn)

3. Код на мові Java:

|  |
| --- |
| **private** **static** **void** mergeSort(**int** a[], **int** l, **int** r) {  **if** (l < r) {  **int** m = (l + r) / 2;  *mergeSort*(a, l, m);  *mergeSort*(a, m + 1, r);  *merge*(a, l, m, r);  }  }    **public** **static** **void** merge(**int** arr[], **int** l, **int** m, **int** r) {  **int** n1 = m - l + 1;  **int** n2 = r - m;  **int**[] leftArray = **new** **int**[n1];  **int**[] rightArray = **new** **int**[n2];  **for** (**int** i = 0; i < leftArray.length; i++)  leftArray[i] = arr[l + i];    **for** (**int** j = 0; j < rightArray.length; j++)  rightArray[j] = arr[m + j + 1];  **for** (**int** k = l, i = 0, j = 0; k <= r; k++) {  **if** (i == leftArray.length) {  arr[k] = rightArray[j++];  **continue**;  }  **if** (j == rightArray.length) {  arr[k] = leftArray[i++];  **continue**;  }  **if** (leftArray[i] <= rightArray[j]) {  arr[k] = leftArray[i++];  } **else** {  arr[k] = rightArray[j++];  }  }  } |

4. Специфікація вхідного набору даних.

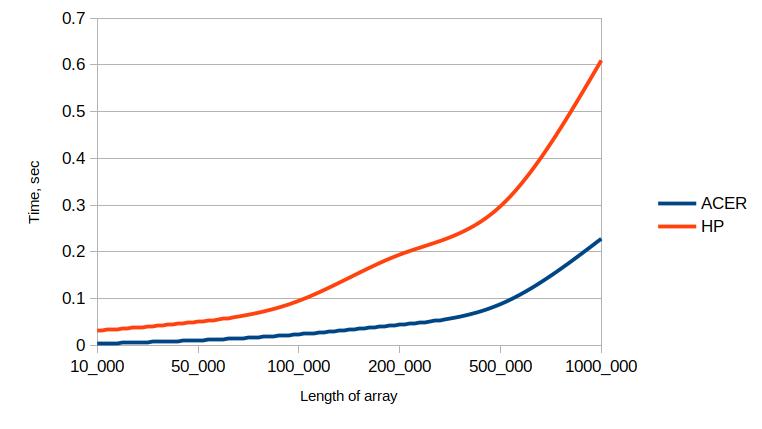
Вхідними даними слугує масив цілих чисел. Масив генерується за домогою методу makeRandom, класу RandomGen.

5. Таблиці з результатами вимірювання:

|  |  |  |
| --- | --- | --- |
| ACER | | |
| # | Довжина масиву | Час, с |
| 1 | 10.000 | 0.00398933 |
| 2 | 50.000 | 0.0112041 |
| 3 | 100.000 | 0.0238112 |
| 4 | 200.000 | 0.0444536 |
| 5 | 500.000 | 0.0884061 |
| 6 | 1.000.000 | 0.228274 |

|  |  |  |
| --- | --- | --- |
| HP | | |
| # | Довжина масиву | Час, с |
| 1 | 10.000 | 0.0318702 |
| 2 | 50.000 | 0.0505193 |
| 3 | 100.000 | 0.0956180 |
| 4 | 200.000 | 0.194290 |
| 5 | 500.000 | 0.297833 |
| 6 | 1.000.000 | 0.609557 |

6. Графіки



7. Висновок

В ході лабораторної роботи емпіричними методами було проведено емпіричний аналіз алгоритму низхідного злиття (Top-Down Merge Sort) на двох апаратних платформах HP X302-UJ та ACER ASPIRE v3-571g. Був виміряний час, що затрачає компьютер при сортуванні масивів цілих чисел різної довжини. Результати представлені в таблиці вимірювань.