|  |
| --- |
|  |
| Извештај – алгоритми за подредување |
| Insertion, selection, merge и quick sort |
|  |
| **Ивано Божиновски** |
|  |

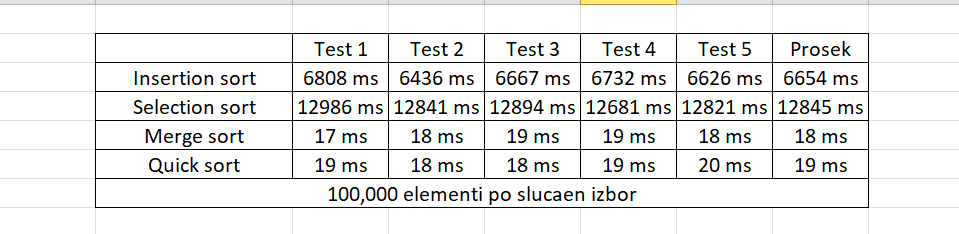
|  |
| --- |
|  |

Во овој извештај се изложени реултатите од експериментот во кој споредувавме 4 алгоритми за подредување: insertion, selection, merge и quick sort.

Сите тестирања се правени на Onlline GDB C++14 компајлер, на лаптоп со следните спецификации:

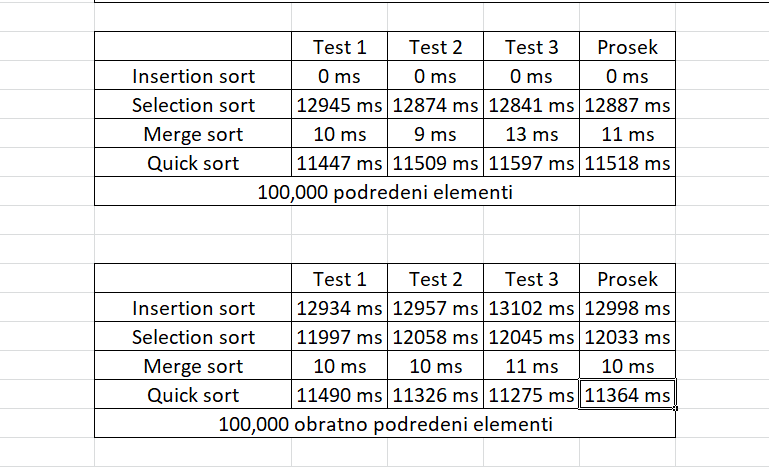
* Intel Core i3-7020U CPU @ 2.30GHz
* 4GB RAM
* Intel HD Graphics 620 (128MB) графичка картичка

Прво, сите алгоритми се пуштени по 5 пати да подредат случајно генерирана низа од 100,000 цели броеви.



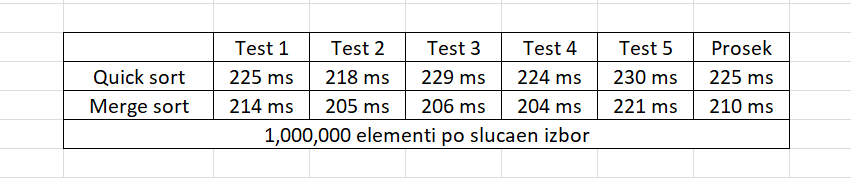
Merge sort и quick sort дадоа многу побрзи резултати од insertion и selection sort, што беше очекувано поради нивните временски комплексности – и , соодветно. Insertion sort е речиси двојно побрз од selection sort бидејќи selection sort мора при секоја итерација да ја изминува низата до крај за да го најде минималниот елемент, а insertion sort го наоѓа точното место за новиот елемент со изминување на половина од низата во просек.

Потоа, алгоритмите се пуштени на подредена и обратно подредена низа од 100,000 цели броеви.



Insertion sort има време од 0 милисекунди на подредена низа бидејќи секогаш го додава најголемиот елемент на крајот од низата. Quick sort е сега многу поспор бидејќи при секоја итерација како елемент за споредување го бира најмалиот/најголемиот број во поднизата, па таа не се дели на половина, туку во еден дел се става само еден елемент, а во друг сите останати. Ако се примени друга имплементација, во која секогаш се бира средишниот или произволен елемент во поднизата, времето на извршување на quick sort ќе биде отприлика исто со она на merge sort за (обратно) подредена низа. За низи генерирани по случаен избор двете имплементации се еднакво брзи, бидејќи шансите да се генерира низа за која алгоритмот нееднакво ќе ги дели поднизите се многу мали.

За крај, двата најбрзи алгоритми ги пуштив по 5 пати на случајно генерирани низи од по 1,000,000 цели броеви.



Тука, merge sort даде малку побрзи резултати.