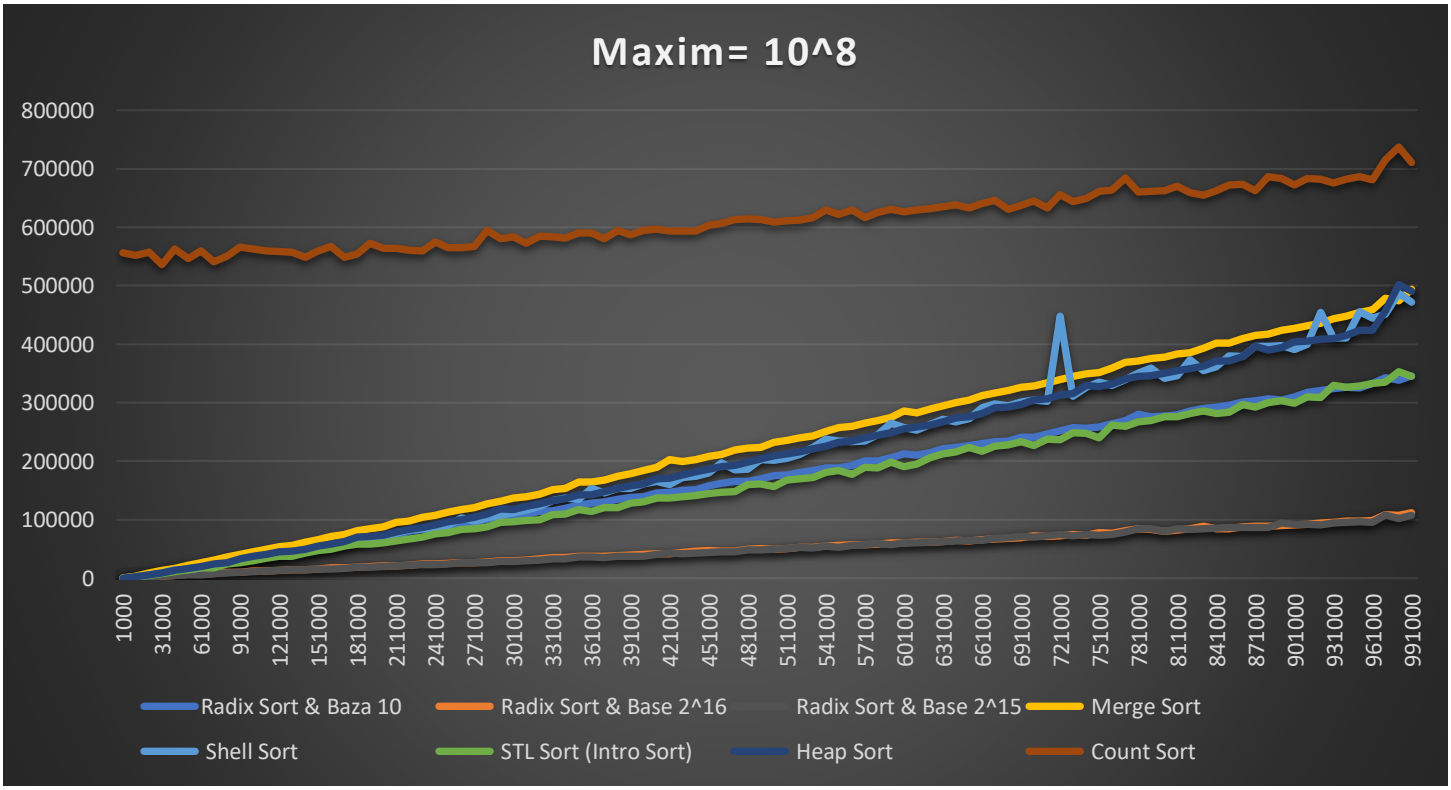
Suport Raport

10^3<=N<=10^6 Maxim=10^8

(**Pe coloana sunt reprezentati timpii de rulare, iar pe linie este reprezentata variatia de dimensiune a vectorului)**



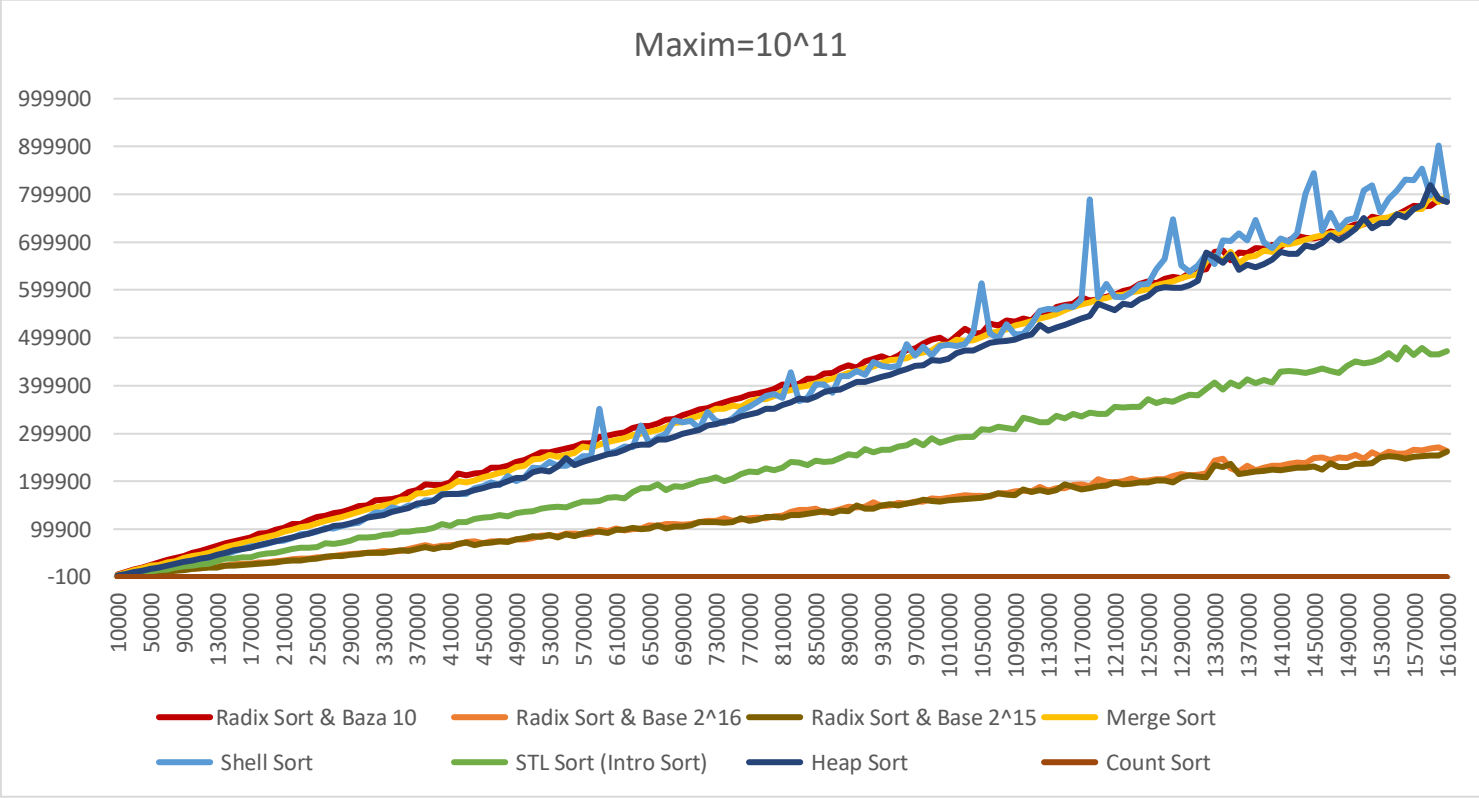
Obs: Totodata, se poate observa diferenta notabila de timpi dintre sortarea Radix Sort, dar cu baze diferite. In timp ce bazele 2^15 si 2^16 se descurca aproximativ la fel, baza 10 (cea standard) cauzeaza o creștere substanțială a timpului de rulare.

OBS: După cum se poate observa, la maxime mari, Count Sort-ul este total ineficent datorita modului lui de functionare pe baza unui vector de frecventa.

Obs: Merge Sort – ul creste liniar in raport cu variabila de dimensiune a vectorului.

Obs: Pe acest set de valori , Sortarea Default din c++(STL Sort/ Intro Sort) se comporta aproximativ la fel cu sortarea Radix Sort cu baza 10.

10^4<=N<=10^8 Maxim=10^11



In codul sursa, am codat eroarea cauzata de memoria alocata in mod excesiv ca timp negativ, astfel se poate observa cum Count Sort-ul esueaza la aceasta valoare a maximului deoarece nu poate sa aloce un vector de 10^11 elemente.

Totodata, se observa cum Shell Sort-ul incepe sa aiba “Spike-uri” la n>=10^7 si Maxim=10^11. In continuare, se observa diferenta notabila de timp dintre acelasi Radix Sort, dar cu baze diferite(fiind vorba de bazele 2^15; 2^16 si baza 10), baza 10 fiind in continuare deficitara din punct de vedere al timpului.

Se observa, de asemenea ca, spre deosebire de cazul anterior unde sortarea STL si sortarea Radix cu baza 10 se comportau aproximativ la fel, aici se comporta diferit, sortarea STL comportandu-se mai bine decat Radix Sort cu baza 10.

Atasat, gasiti raporturile csv corespunzatoare graficelor din acest document.