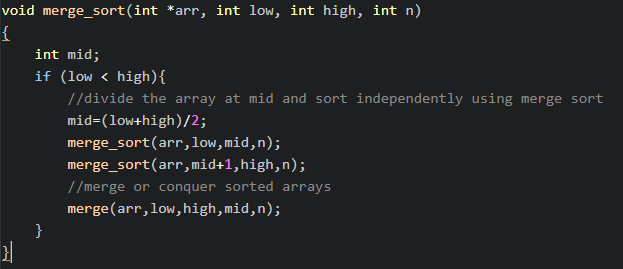
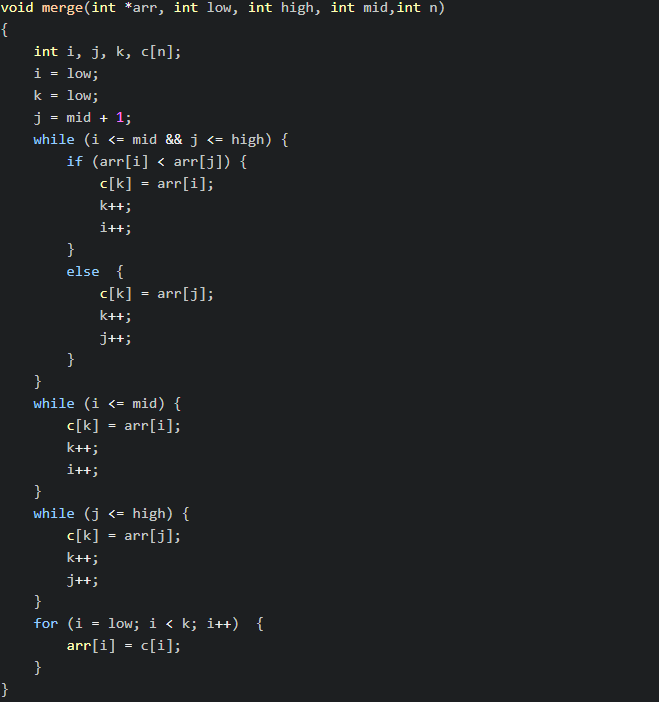
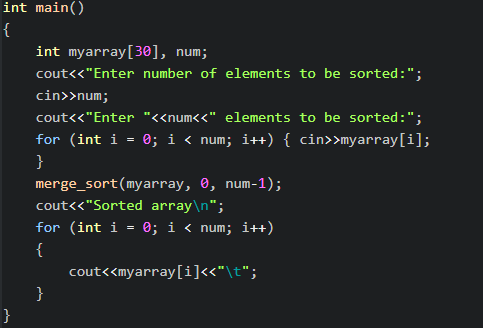
قسمت 1)

در این قسمت ابتدا تمام الگوریتم های سورت را می نویسیم سپس برای هر یک، یک main نوشته و با اجرای ان فایل الگوریتم را تست می کنیم:

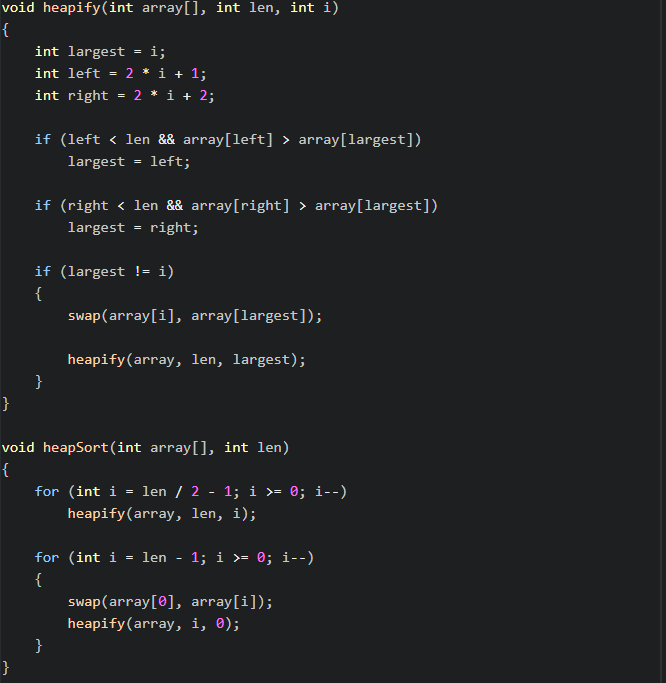
1. Merge Sort:

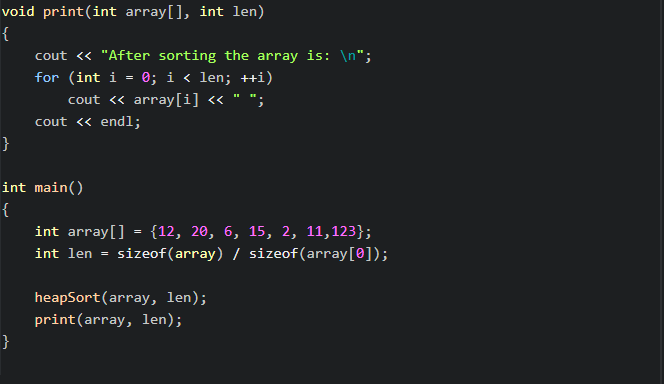




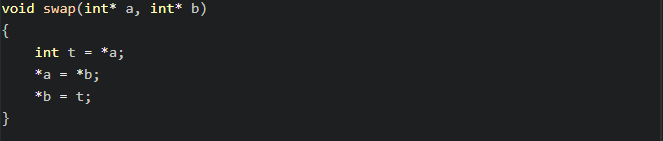


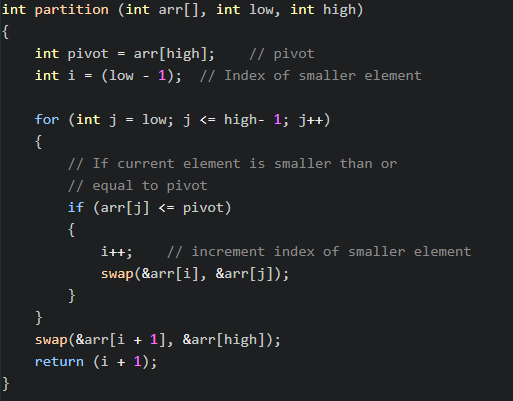
1. Heap Sort:

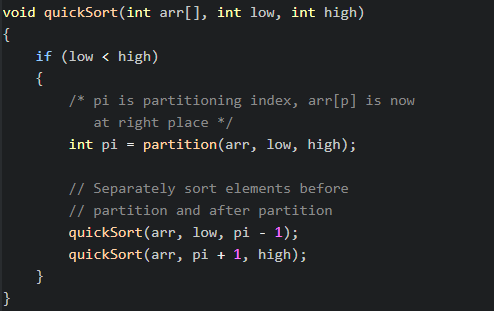


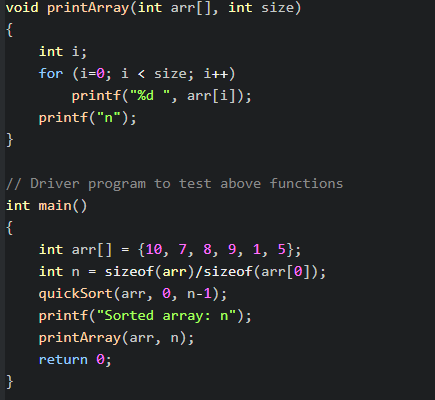


1. Quick Sort:

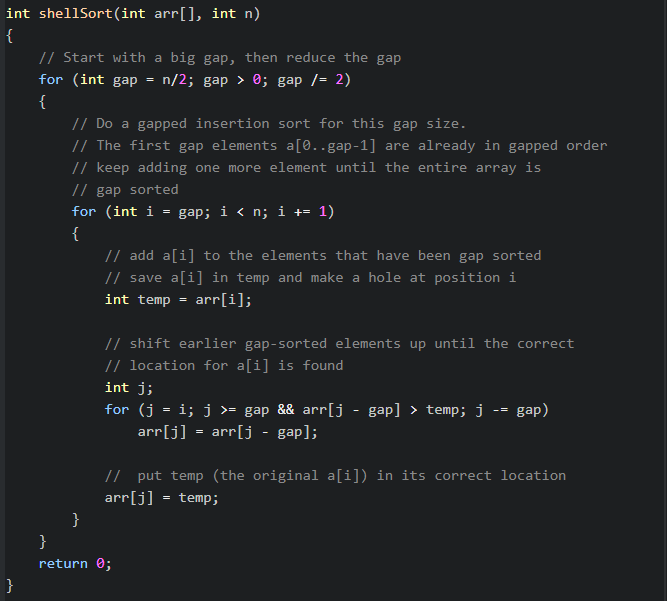


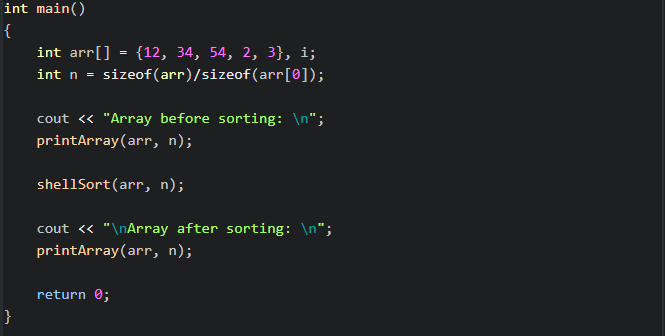




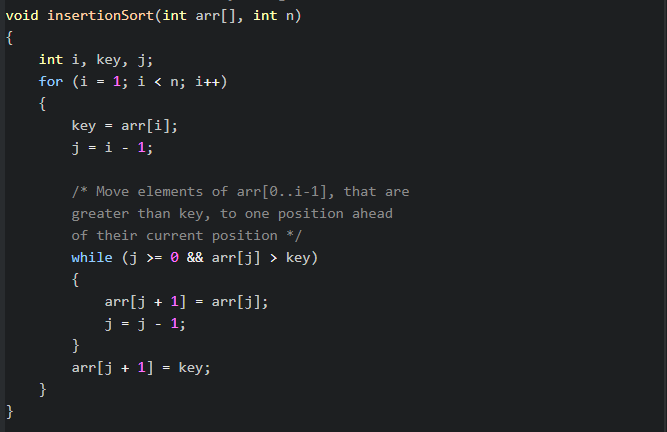


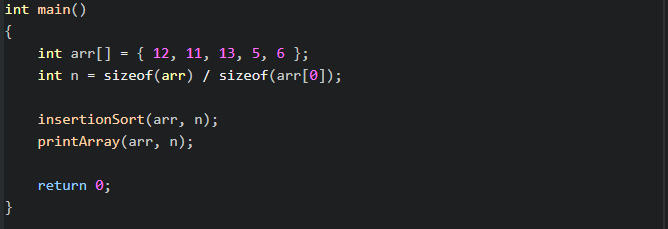
1. Shell Sort:





1. Insertion Sort:

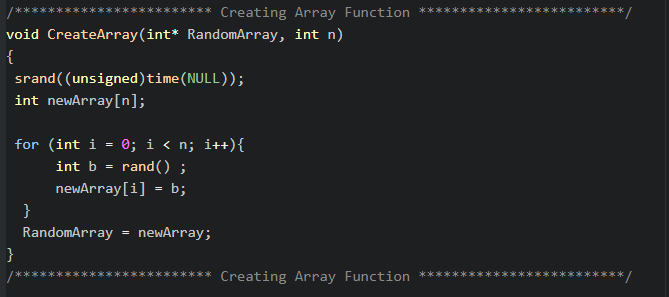




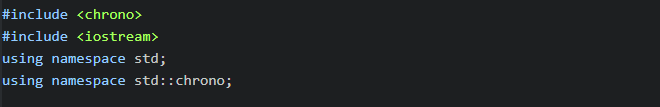
قسمت 2)

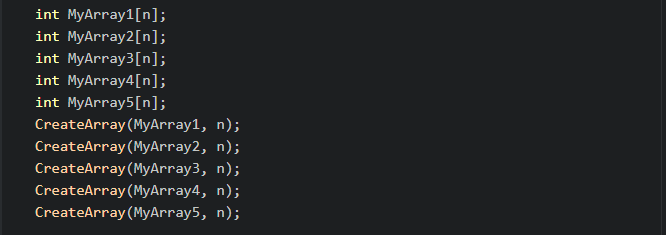
سپس در این قسمت تمام سورت ها را در یک فایل قرار می دهیم تا آنها را به ترتیب تست کنیم.

بعد از ان یک تابع می نویسیم تا با گرفتن یک عدد ورودی یک ارایه Uniform با ان تعداد عضو بسازد پس داریم:

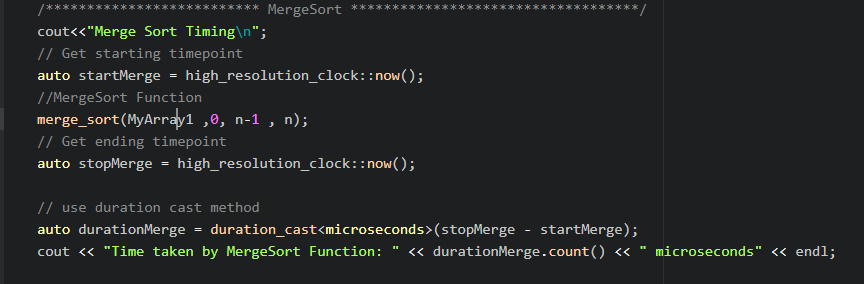


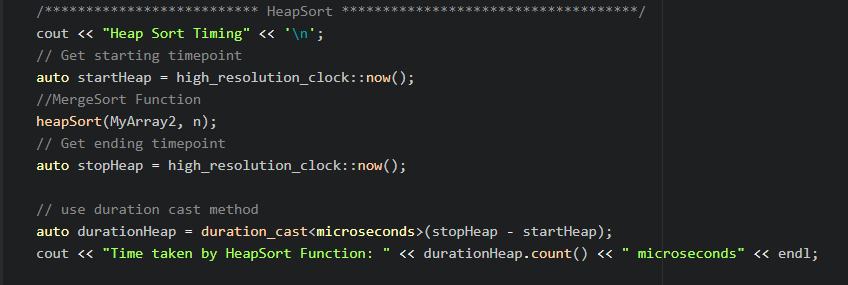
سپس یک main نوشته و در ان 5 ارایه با استفاده از تابع ساخته شده می سازیم و ان ها را به عنوان ورودی تابع های سورت می گذاریم.

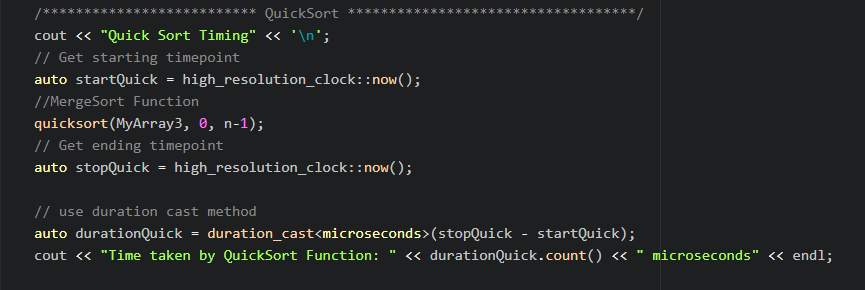
برای بدست اوردن زمان اجرای هر دستور از کتابخانه ی 

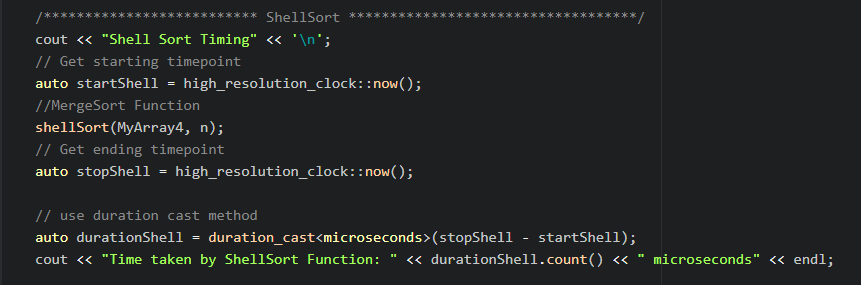
Chrono استفاده کرده ایم. 

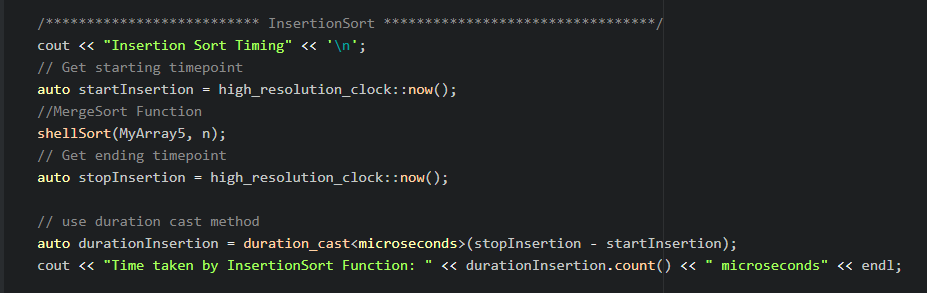
سپس برای هر کدام از سورت ها این ورودی ها را می دهیم پس داریم:





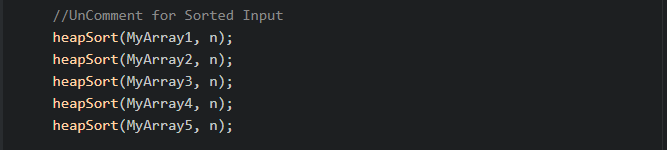




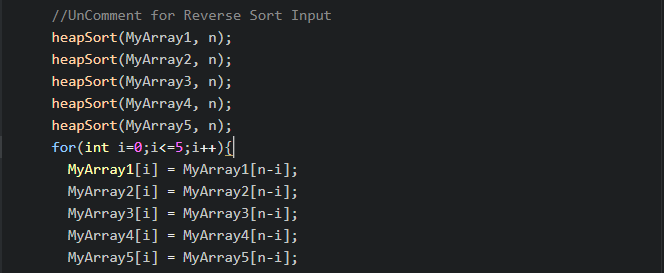


در این حالت زمان اجرای هر یک را بدست می آوریم.

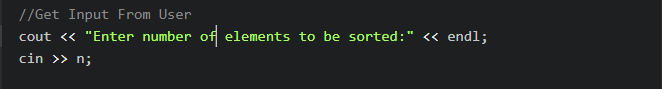
برای حالت ورودی سورت شده ابتدا ارایه ی ساخته شده را سورت می کنیم و سپس ان را به عنوان ورودی به تابع های سورت می دهیم:



برای حالت ورودی Reverse Sort مانند زیر ابتدا ورودی ها را سورت کرده سپس ارایه را بر عکس می کنیم و سپس ان ها را به عنوان ورودی به تابع های سورت می دهیم:



برای گرفتن تعداد عضو های ارایه از ورودی نیز مانند زیر عمل می کنیم:



و از n به عنوان تعداد عضو های ارایه استفاده می کنیم

بعد از اجرای کد زمان اجرا باری حالات مختلف به شکل زیر است:

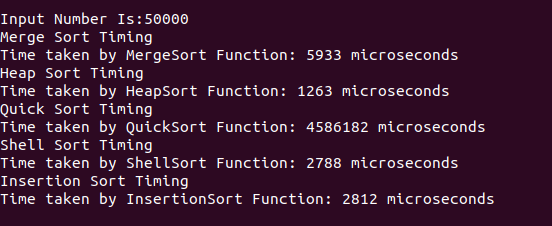


Figure :50K Normal

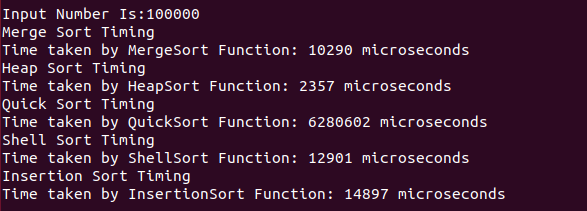


Figure :100K Normal

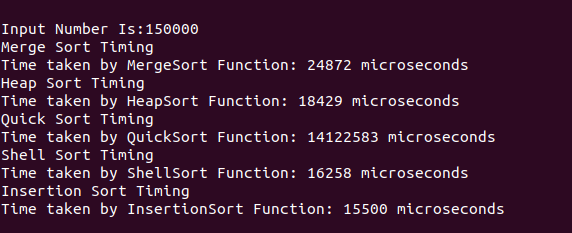


Figure :150K Normal

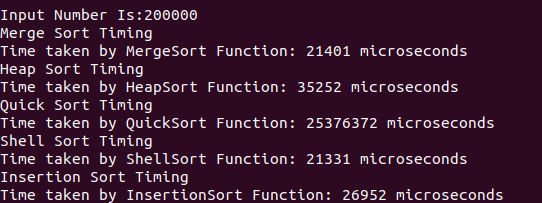


Figure :200K Normal

برای حالت ورودی سورت شده داریم:

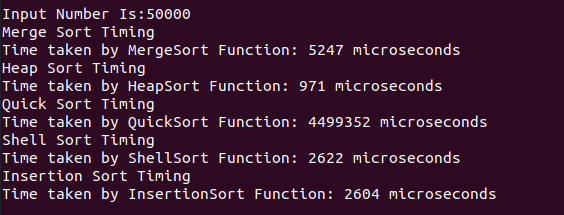


Figure :50K Sorted

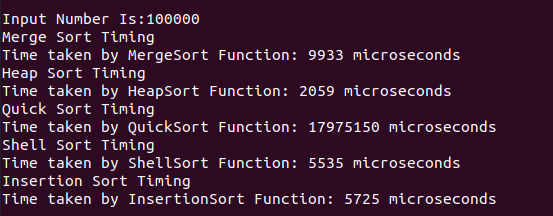


Figure :100K Sorted

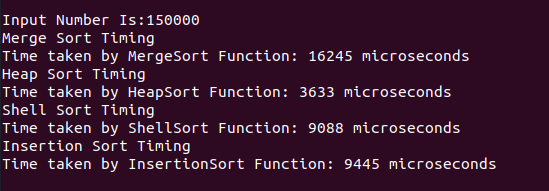


Figure :150K Sorted

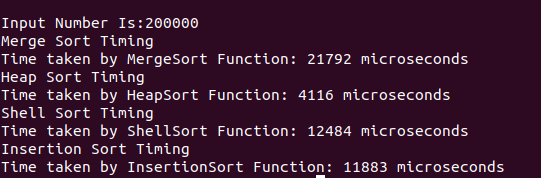


Figure :200K Sorted

برای حالت ورودی سورت شده بر عکس داریم:

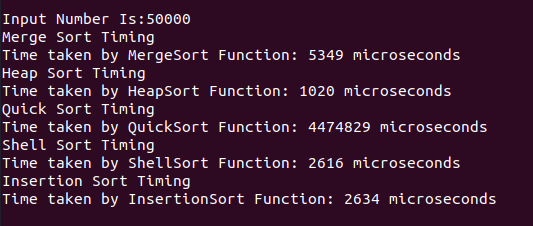


Figure :50K Reverse

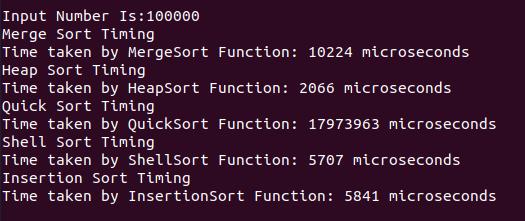


Figure :100K Reverse

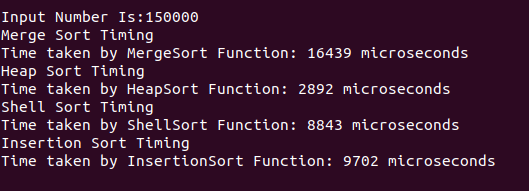


Figure :150K Reverse

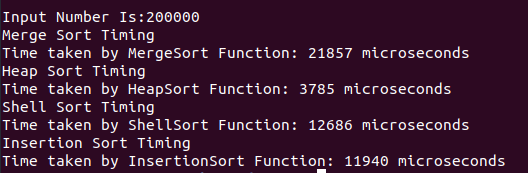


Figure :200K Reverse

بعد از اجرای کد ها برای زمان اجرای آنها جدول های زیر را داریم:

Sorted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 50K Input | 100K Input | 150K Input | 200K Input |
| Merge Sort | 5247 | 9933 | 16245 | 21792 |
| Heap Sort | 971 | 2059 | 3633 | 4116 |
| Quick Sort | 4499352 | 17975150 |  |  |
| Shell Sort | 2622 | 5535 | 9088 | 12484 |
| Insertion Sort | 2604 | 5725 | 9445 | 11883 |

Reverse Sort

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 50K Input | 100K Input | 150K Input | 200K Input |
| Merge Sort | 5312 | 10224 | 16439 | 21857 |
| Heap Sort | 976 | 2066 | 2892 | 3785 |
| Quick Sort | 4474829 | 17973963 |  |  |
| Shell Sort | 2719 | 5707 | 8843 | 12686 |
| Insertion Sort | 2705 | 5841 | 9702 | 11940 |

Normal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 50K Input | 100K Input | 150K Input | 200K Input |
| Merge Sort | 5933 | 10290 | 24872 | 21401 |
| Heap Sort | 1263 | 2357 | 18429 | 35252 |
| Quick Sort | 4586182 | 6280602 | 14122583 | 25376372 |
| Shell Sort | 2788 | 12901 | 16258 | 21331 |
| Insertion Sort | 2812 | 14897 | 15500 | 26952 |

برای درجه هر کدام از Sort ها داریم:

Curve Fitting

در این قسمت اطلاعات در جدول را به Matlab می دهیم و جدول زیر را بدست می آوریم: