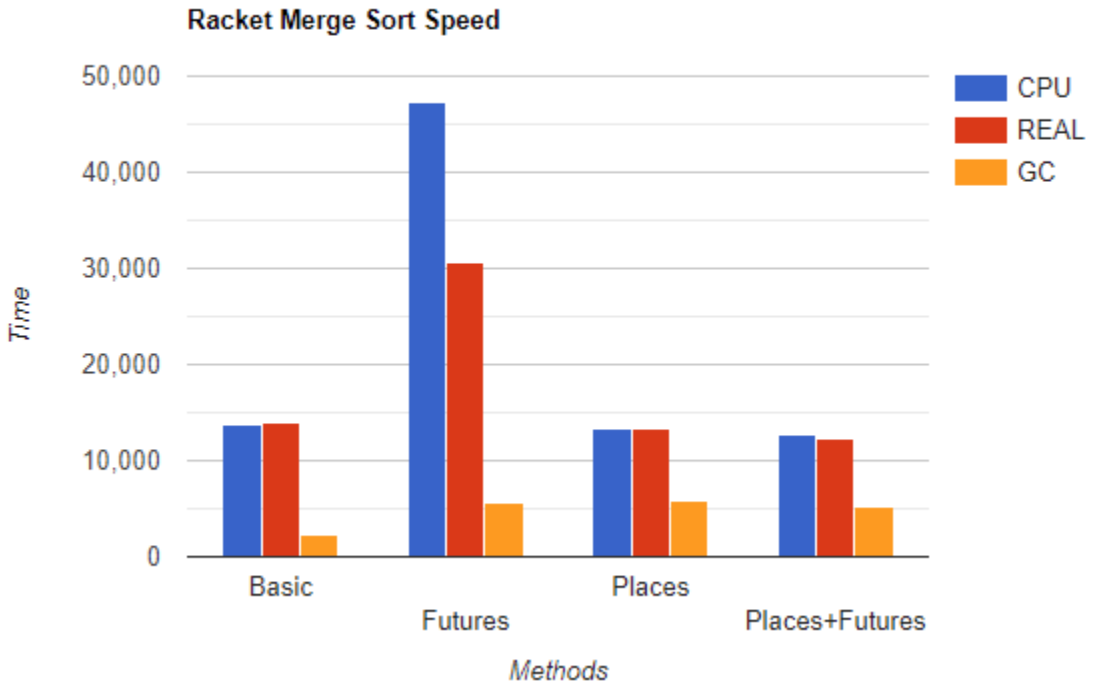
Racket Programming Assignment
Connor Schafer
CS441

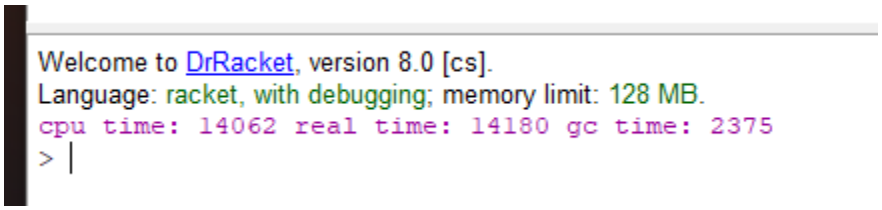


The results for this assignment were overall very surprising. My fastest run times in terms of CPU and Real time ended up being the method with Places+Futures, which is shocking due to each method individually being worse, places being slightly worse, and futures being awful. I think the fact that with or without adding parallelism got relatively the same answer is due to the fact that merge sort is a fairly simple program and so the upfront cost of creating methods outways the benefit. On an even larger file, or a higher scaled program you can clearly see it will pay off (maybe not futures). Another possibility of the parallelism performance could be my method of implementation and not being able to optimize these methods.

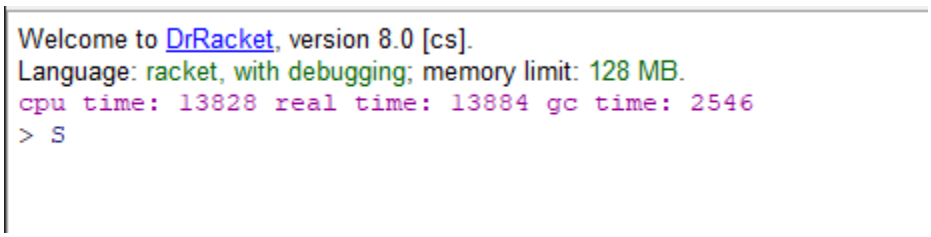
Overall this assignment was a big challenge, and I think that is due to the lack of resources available. Learning the syntax to racket is a hurdle, but as you work your way through the assignment, it slowly starts to make more sense.

If you are curious, below I have more specifics on each run.

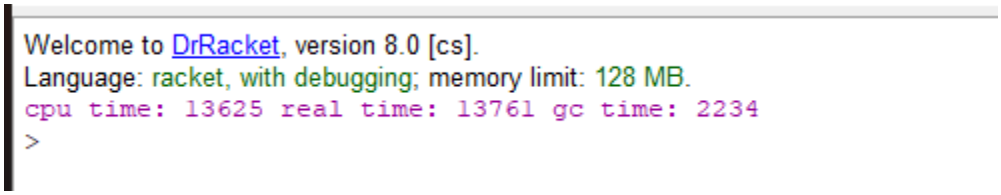
Run with full text file, no parallelism, 128MB memory limit,



2nd

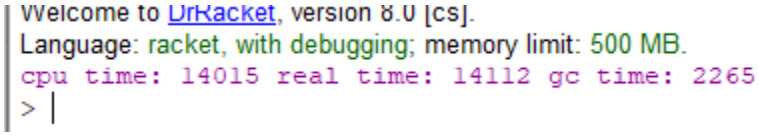


3rd



AVERAGE CPU TIME: 13,838
AVERAGE REAL TIME: 13,941
AVERAGE GC TIME: 2,385

For testing purposes I increased, the memory limit to 500MB to see if this would increase performance, still with no parallelism



The Results were inconclusive.

Run with full text file, with futures, 2000MB memory limit,
1st

```
Welcome to DrRacket, version 8.0 [cs].
Language: racket, with debugging; memory limit: 2000 MB.
cpu time: 48125 real time: 30509 gc time: 6234
>
```

2nd

```
Welcome to DrRacket, version 8.0 [cs].
Language: racket, with debugging; memory limit: 2000 MB.
cpu time: 44843 real time: 29891 gc time: 6218
>
```

3rd

```
Welcome to DrRacket, version 8.0 [cs].
Language: racket, with debugging; memory limit: 2000 MB.
cpu time: 49015 real time: 31308 gc time: 4343
>
```

AVERAGE CPU TIME: 47,327
AVERAGE REAL TIME: 30,569
AVERAGE GC TIME: 5,598

Run with full text file, with places, 2000MB memory limit,

1st

```
Welcome to DrRacket, version 8.0 [cs].
Language: racket, with debugging; memory limit: 2000 MB.
> (main)
cpu time: 14515 real time: 14104 gc time: 6640
>
```

2nd

```
Welcome to DrRacket, version 8.0 [cs].
Language: racket, with debugging; memory limit: 2000 MB.
> (main)
cpu time: 13046 real time: 12835 gc time: 5203
>
```

3rd

```
Language: racket, with debugging; memory limit: 2000 MB.
> (main)
cpu time: 13453 real time: 13212 gc time: 5515
>
```

AVERAGE CPU TIME: 13,383
AVERAGE REAL TIME: 13,383
AVERAGE GC TIME: 5,786

Run with full text file, with places AND futures, 2000MB memory limit,

1st

```
Language: racket, with debugging; memory limit: 2000 MB.
> (main)
cpu time: 12578 real time: 12227 gc time: 5234
>
```

2nd

```
Welcome to DrRacket, version 8.0 [cs].
Language: racket, with debugging; memory limit: 2000 MB.
> (main)
cpu time: 12843 real time: 12241 gc time: 5281
>
```

3rd

```
Language: racket, with debugging; memory limit: 2000 MB.
> (main)
cpu time: 12859 real time: 12422 gc time: 5125
>
```

AVERAGE CPU TIME: 12,760
AVERAGE REAL TIME: 12,296
AVERAGE GC TIME: 5,213
