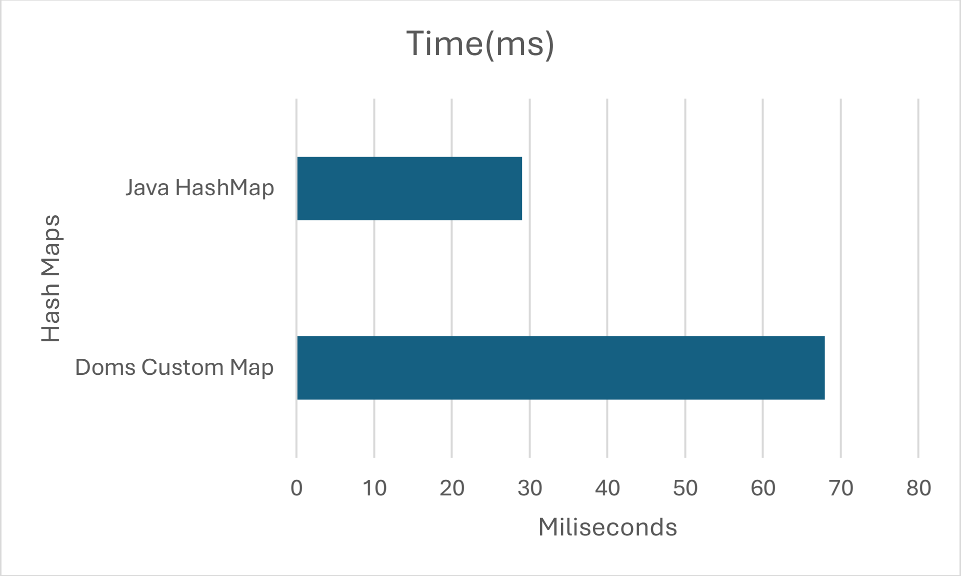
Project 2 Part 3

The point of Part 3 was to create a hash map and run an experiment by running a large dataset through it and recording anything interesting. I built off of the dumbHashMap that we created in class. I also decided to use the Pokemon CSV dataset that I used for Part 2 to run my experiment.

- **DominicZelinskyHashMap.java|**src\main\java\Part3_DominicZelinskysHashMapFolder\DominicZelinskyHashMap.java
  - ***dumbHash():*** This method used to sum upthe ASCII value for whatever the inputed string is
  - ***contains():*** This method is used to check whether or not the item is already within the bucket
  - ***resize():*** This method is used to resize the hash map, to make it larger for adding more items
  - ***size():*** This method is used to calculate the size of the HashMap, iterating through all of the buckets and summing them all together to get the true size
  - ***put():*** This method is used to add more values into the HashMap
  - ***get():*** This method is used to get a value from the HashMap.
  - ***remove():*** This method is used to remove a value from the HashMap
  - ***getCollisions():*** This gets the collision counter integer
- **PokemonReader.java** | File Path: src\main\java\Part3_DominicZelinskysHashMapFolder\PokemonReader.java
  - ***readPokemonCSV():*** This method is adapted from the DataHandler.java class in order to specifically adapt the Pokemon TCG CSV into an arraylist to add into my Hash Map
- **PokemonHasher.java** | File Path: src\main\java\Part3_DominicZelinskysHashMapFolder\PokemonHasher.java
  - ***hashPokemon():*** This method resizes my hashmap to the size ofhte pokemon arraylist. It then iterates through the pokemon arraylist, condenses the important data into one string and adds it into my hashmap.
- **HashRace.java** | File Path: src\main\java\Part3_DominicZelinskysHashMapFolder\HashRace.java
  - ***hashRace():*** This method is used to compare Java Standard HashMap speeds and my Custom Hashmaps speeds. It does this by timing how long it takes both HashMaps to add the Pokemon CSV into the respective Hashmaps. This then calls the raceResults() method from RaceCSVOutput.java to record the race results.

- **RaceCSVOutput.java** | File Path: src\main\java\Part3_DominicZelinskysHashMapFolder\RaceCSVOutput.java
    - ***raceResults():*** This method intakes the time results from the HashRace.java class and outputs it into a CSV for graphing purposes.



I thought that it was interesting that despite my hash map being simpler, and not requiring 2 values to be stored, that the Java HashMap was more than twice as fast at adding the Pokemon CSV into it. I also thought it was interesting that when adding these items to my hash map, that I had encountered 19597 collisions.