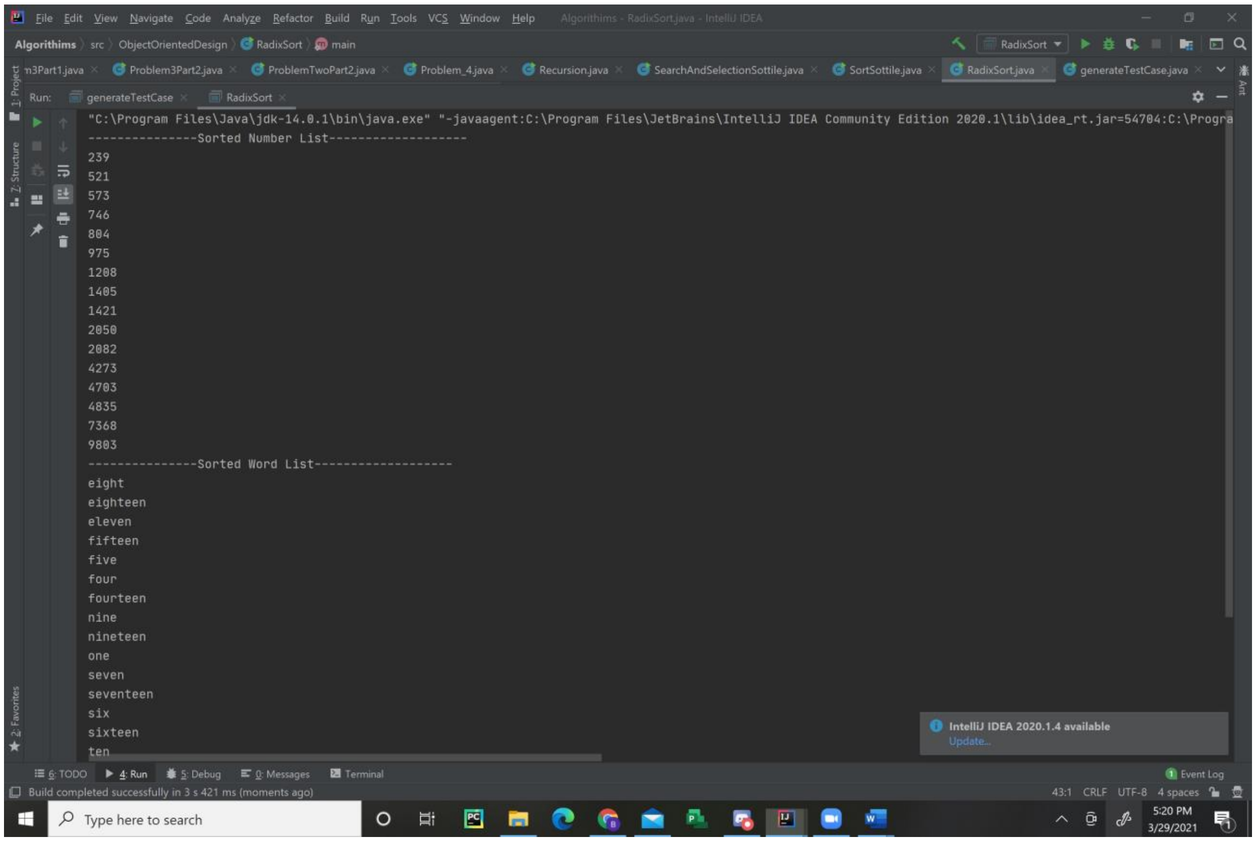
Ben Sottile

CS 310

Radix Sort

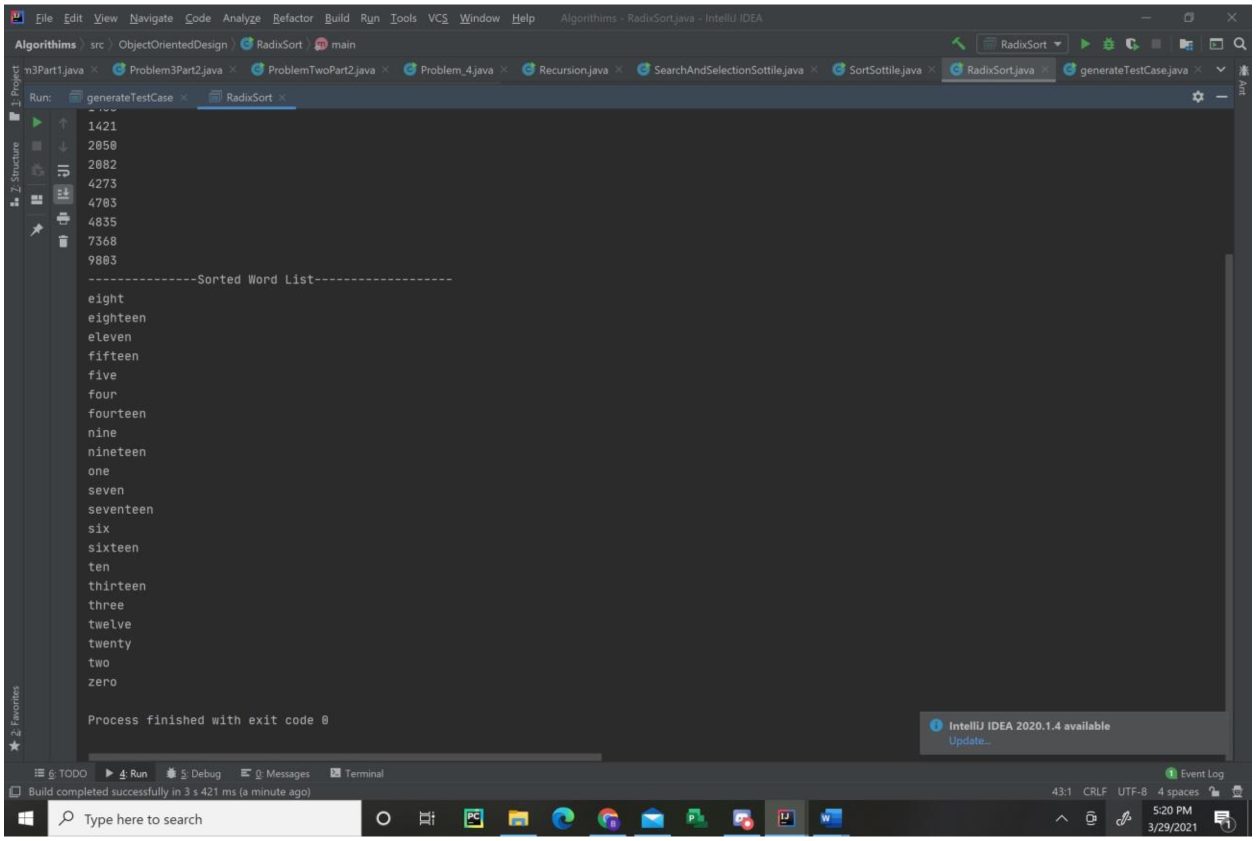
Report

Screenshots of Running Code:


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Algorithims - RadixSort.java - IntelliJ IDEA
Algorithims src ObjectOrientedDesign RadixSort main
n3Part1.java Problem3Part2.java ProblemTwoPart2.java Problem_4.java Recursion.java SearchAndSelectionSottile.java SortSottile.java RadixSort.java generateTestCase.java
Run: generateTestCase RadixSort
IntelliJ IDEA 2020.1.4 available
Update...
6: TODO 4: Run 5: Debug 0: Messages Terminal Event Log
43:1 CRLF UTF-8 4 spaces
Type here to search
5:20 PM 3/29/2021

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.1\lib\idea_rt.jar=54704:C:\Progra
--------------Sorted Number List------------------
239
521
573
746
804
975
1208
1405
1421
2050
2082
4273
4703
4835
7368
9803
--------------Sorted Word List------------------
eight
eighteen
eleven
fifteen
five
four
fourteen
nine
nineteen
one
seven
seventeen
six
sixteen
ten
```

Build completed successfully in 3 s 421 ms (moments ago)



```
1421
2050
2082
4273
4703
4835
7368
9803
---------------Sorted Word List-------------------
eight
eighteen
eleven
fifteen
five
four
fourteen
nine
nineteen
one
seven
seventeen
six
sixteen
ten
thirteen
three
twelve
twenty
two
zero

Process finished with exit code 0
```

Report:

I have managed to successfully code the entirety of the Radix sort for sorting ints from smallest to Largest and for alphabetizing strings. This process was a long on. For the integer radix sort I had to figure out my plan for the append() function and the combine buckets function in the pseudo code. Once I figured out a way I built the frame work from the psudo code to create the full function. My only difficulty here was human error as I had not cleared my buckets after going through each digit. The result was an overwritten list that caused index out of bounds errors and confounded me for a good while. Once I reached the alphabetizing assignment I created the my normalize and weirdize functions to append filler characters(I used a *) to ensure the strings were all of a uniform format. I also made a function to figure out the size of the largest string. I then used the same frame work from the integer assignment which sent me to a very long trial and error process. My first error was in the fact that the string without a * character was creating an error in my weirdize function because it would only stop check once it reached a *. Since there was no * it threw an error. After that I had to deal with figuring out the best way to correctly alphabetize using a radix sort. The variables I keep changing until I got a working answer(not the best idea) were the location of the filler characters(before vs after), how I read the string (forward to end vs end to forward), and the later discovered third category(priority of the * characters vs letters. The final code puts star highest priority, put the filler characters as the back of the string and reads it backwards. I verified my answers through randomly generated testcases from random word generator.com(I copy and pasted them into a program I made that add ,'s and ",s so I could copy and paste them into the list.

Radix Sort

Chapter 4 of the textbook discusses various algorithms for sorting data, one of which is Radix Sort. Along with the algorithm description and pseudo code presented are various analyses (best case, worst case, average case) for the algorithm with the big-oh complexities. In this assignment you will implement the algorithm to perform the sort but you will not be asked to empirically verify the analysis.

Assignment

Radix Sort – Numeric data

Implement the Radix Sort algorithm based on the pseudo code given on page 95 of the textbook, sorting a list of integer values from lowest to highest. Demonstrate the code by running it on the list of values given in Exercise 1, section 4.2.2 Exercises on page 98 of the text. Print the sorted list to demonstrate correctness.

Radix Sort – String data

Implement the Radix Sort algorithm based on the pseudo code given on page 95 of the textbook, sorting a list of String values alphabetically. Note that this can be a completely separate implementation from the version above. You may assume that the String values are all lower case. Demonstrate the code by running it on this list of String values:

```
String[] unsorted = {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight",
                     "nine", "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen",
                     "sixteen", "seventeen", "eighteen", "nineteen", "twenty"};
```

Print the sorted list to demonstrate correctness.

Deliverables

- Source code
- Essay describing successes, difficulties, and a brief description of the results and screen shots of the running programs. This document must be submitted in PDF format.