

# **Programming Assignment 2: Social Security Number Sorter**

Jen Luu

CS 146

Professor Wu

April 28, 2019

## **A. Design and Implementation**

### **a. Data structures**

- i. The primary data structures used in the social security number sorter were integer and String Arrays. Both the integer and String Arrays contained 300 social security numbers at all times. A dynamic data structure was not needed because size was initialized to 300 at the beginning and was fixed. String Arrays stored the String version of the social security numbers, which included the dashes between the different partitions. The String Arrays were used for the printing process as well. On the other hand, integer Arrays stored the integer value of each social security number. The dashes between the partitions were removed in order for the numbers to be compatible with the arguments of the sorting algorithms.

### **b. Radix sort: defining the digit**

- i. Radix sort is based on sorting from the least significant digit (LSD) to the most significant digit (MSD) of a number. Therefore, each iteration of Radix sort requires specification of which digit to start with. The modulus operation (%) is required in the implementation of Radix sort because it performs integer division. Hence, the remainder will be a digit.
- ii. As Radix sort iterates from the LSD to the MSD, there must be a way to specify which digit to perform the sorting operation on. Therefore, each

number must be divided by a power of 10 as Radix sort moves from right to left.

iii. Example:

Obtain the LSD:

$$10^0 = 1$$

$$123456789 / 1 = 123456789$$

$$123456789 \% 10 = 9$$

Next digit:

$$10^1 = 10$$

$$123456789 / 10 = 12345678$$

$$12345678 \% 10 = 8$$

Next digit:

$$10^2 = 100$$

$$123456789 / 100 = 1234567$$

$$1234567 \% 10 = 7$$

.

.

.

MSD:

$$10^8 = 100,000,000$$

$$123456789 / 100,000,000 = 1$$

1.  $1 \% 10 = 1$

c. Bucket sort: designing a bucket

- i. Bucket sort relies on having multiple buckets to distribute the elements of an Array. In the social security number sorter, a main `ArrayList<Integer>[]` is initialized to hold 10 `ArrayLists<Integer>`. To group the number in the appropriate bucket, the number is first divided by 100,000,000 to obtain a single digit. However, the original number will

not be altered. The digit will be the sorting condition for which bucket the number will belong to. After all the numbers have been distributed, insertion sort will sort each bucket and the results will be concatenated together.

ii. Example:

1. SSN: 112348592, 362128912, 634129031

	[0, 1)	0	⇒	/
	[1, 2)	1	⇒	112348592
	[2, 3)	2	⇒	/
	[3, 4)	3	⇒	362128912
	[4, 5)	4	⇒	/
	[5, 6)	5	⇒	/
	[6, 7)	6	⇒	634129031
	[7, 8)	7	⇒	/
	[8, 9)	8	⇒	/
2.	[9, 10)	9	⇒	/

## B. Classes and Methods

### a. Number

i. `public Number()`

1. The constructor for the class randomly generates a social security number by calling on `(int)(Math.random()*900 + 100)` for the area number, `(int)(Math.random()*90 + 10)` for the group number, and `(int)(Math.random()*9000 + 1000)` for the serial number. The private global String `ssNumber` is set by adding the area, number, and serial parts together and separating them with a dash (-).
- ii. `public String getSSN()`
  1. This is a getter method that returns the social security number as a String.

#### **b. QuickSort**

- i. `public static void quicksort(int[] A, int p, int r)`
  1. The arguments take in an unsorted `int[]`, the first index of the Array `p`, and the index of the pivot `r`. Before the method proceeds, `p` must be less than `r` in order for the the sorting to work. Afterwards, `A` is partitioned into two Arrays and quicksort is recursively called on both of them until the full Array is sorted in ascending order.
- ii. `public static int partition(int[] A, int p, int r)`
  1. The arguments take in an unsorted `int[]`, the first index `p`, and the pivot `r`. Two points are initialized, `i` and `j`. The method will iterate through the Array in a for loop, where `i = p - 1` and `j = p`. `A[j]` will be compared with `A[r]`. If `A[j] <= A[r]`, `i` will increment and `A[i]` and `A[j]` will swap. If `A[j] > A[r]`, then nothing else will happen

and j will increment. This process will continue until  $j = r$ . At that point, the for loop will terminate and  $A[i+1]$  and  $A[r]$  will swap.

The method will return  $q = q + 1$ , where q will be used to partition the left and right subarrays.

### c. RadixSort

i. `public static void countingSort(int[] A, int[] B, int d)`

1. The arguments will take in an unsorted `int[] A`, an allocated `int[] B` to hold the output, and `int d` to specify the digit to be sorted. It is important to note that `d` is expected to be some power of 10.
2. To begin, a new `int[] C` of size 10 will be created to hold the occurrences of the digits. Each digit may be any value from 0 to 9; therefore, the size of `C` is in accordance to base 10.
3. In a for loop, `C[i]` will be initialized to 0 to prepare for the counting.
4. In another for loop,  $(A[j]/d) \% 10$  will obtain the digit and `C[i]` will increment. Afterwards, each `C[i]` will add the number of its own occurrences and the occurrences of `C[i - 1]`.
5. The sorting will take place, where `C[i]` will determine the position of the element in `A` will be placed in `B`. Each time `C[i]` is called, the number of occurrences will decrement to prepare for next call.
6. Once `B` is filled with the sorted elements based on digit `d`, the method will update `A`.

ii. `public static void radixSort(int[] A, int d)`

1. The arguments are an unsorted `int[] A`, and the highest-order digit `d`.
2. Before the method calls Counting sort, an auxiliary Array B is initialized with the size of A. The length of the iteration for Counting sort to be called on will be `Math.pow(10, d)` times, where each increment is a power of 10. This is needed to specify which digit Counting sort will be used on. The immediate sorting will occur until the length of A has been reached.

#### **d. BucketSort**

i. `public static void insertionSort(ArrayList<Integer> A)`

1. The argument is an unsorted `ArrayList<Integer> A`. Insertion sort will begin a for loop with the second element and continue until the entire list has been reached. An inner for loop will be used to insert the current value into the appropriate place in the sorted sequence.

ii. `public static void bucketSort(int[] A)`

1. This method will take in an unsorted `int[] A`. It will begin by creating a bucket that contains 10 lists for 10 subintervals. The method will go through each element in A and divide it by 100,000,000 to obtain a single digit. The digit will be the indicator to which bucket the original element will be placed in. Afterwards,

Insertion sort will be called to sort each individual bucket and the results will be concatenated. A will be updated based on the sequence of the lists.

**e. Sorter**

i. `public static void convertToString(int[] A)`

1. The input is an `int[] A` of the social security numbers, and the method will take each element in `A` and convert it to a `String`. The results will be stored in a temporary `String[]`.

ii. `public static void sortRegions(String[] A, String title)`

1. The method will take in a `String[] A`, and a `String` title for the name of the file that will be written with a `Buffered Writer`. The counters for each region in the United States will be initialized to 0. The method will use a for loop to iterate through `A` and obtain the first three digits of each element. The `String` will be converted into an `int`, which will be used to increment the appropriate counter.
2. Once that is finished, the `Buffered Writer` will write the sorted sequence of social security numbers and the population for each region.

iii. `public static void useQuicksort(int[] A)`

1. This will create a `Quicksort` object and call the respective method to sort `A`.



- iv. `public static void useRadixSort(int[] A, int d)`
  - 1. This will create a Radix sort object and call the respective method to sort A based on the highest-order digit d.
- v. `public static void useBucketSort(int[] A)`
  - 1. This will create a Bucket sort object and the respective method to sort A.
- vi. `public static void convertToInt(String[] A, int[] B)`
  - 1. Given a `String[] A`, the method will create an integer without the dashes in the social security number and store it into `int[] B`.
- vii. `public static void writeSSN(String[] A, String title)`
  - 1. The arguments are a `String[]` and the String title of the text file.  
  
The method will use a Buffered Writer to record and number the elements of A.

### C. Self-testing Screenshots

- a. Generate 300 random social security numbers:

- i. Code

```
public Number()
{
    //Multiply by 900 to get the range from 0 - 899 and add by 100 to shift the range from 100 - 999
    int area = (int)(Math.random()*900 + 100);
    //Multiply by 90 to get the range from 0 - 89 and add by 10 to shift the range from 10 - 99
    int group = (int)(Math.random()*90 + 10);
    //Multiply by 9000 to get the range from 0 - 8999 and add by 1000 to shift the range from 1000 - 9999
    int serial = (int)(Math.random()*9000 + 1000);

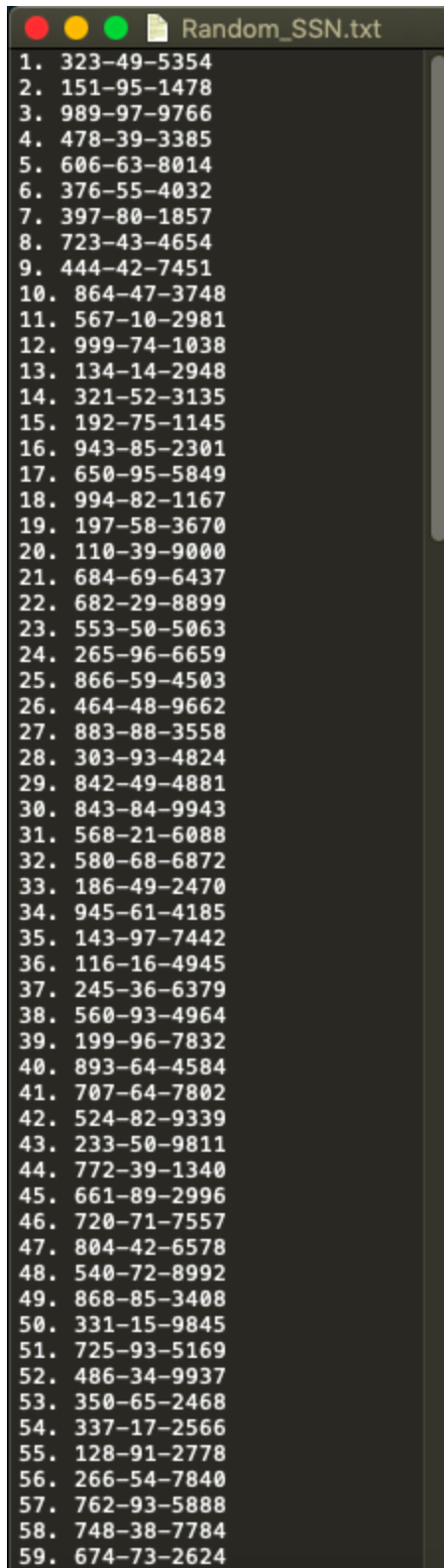
    this.ssNumber = area + "-" + group + "-" + serial;
}
1.
```

- ii. Input

```
Welcome to the social security number sorter.  
Please enter '1' to generate 300 random social security numbers and write them into a text file.  
1  
The process has been completed. Please search your device with the label 'Random_SNN.txt' to see the file.
```

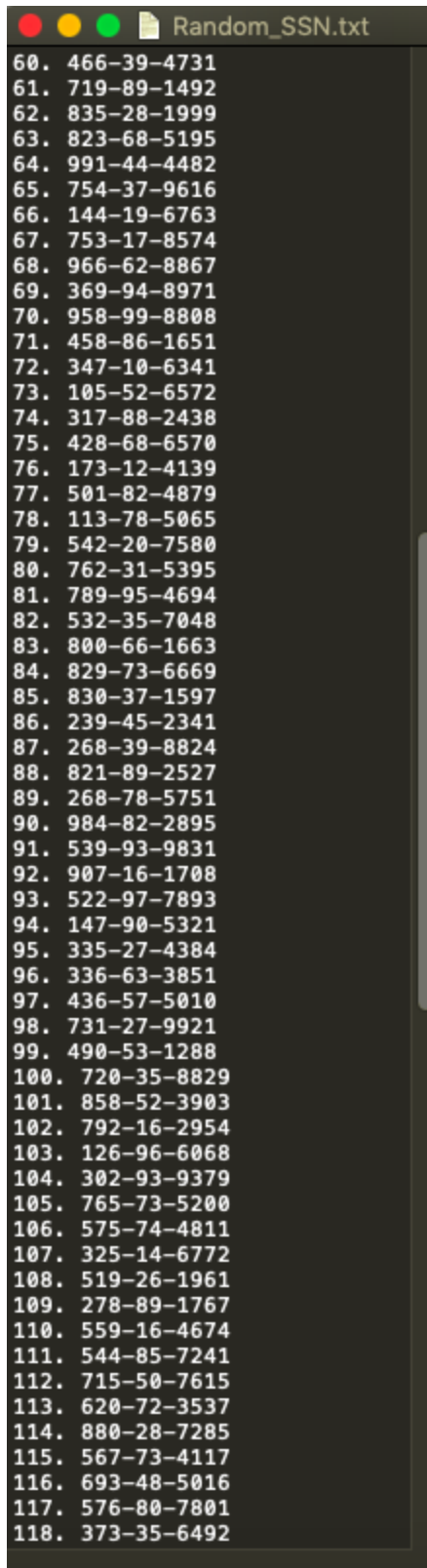
1. -----

### iii. Output



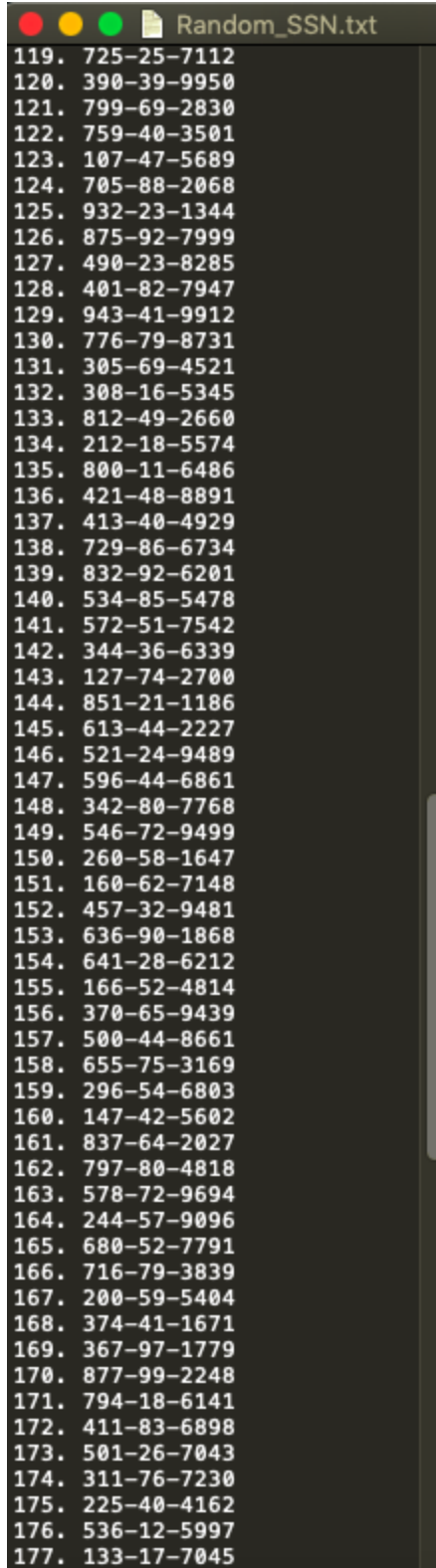
1. 323-49-5354  
2. 151-95-1478  
3. 989-97-9766  
4. 478-39-3385  
5. 606-63-8014  
6. 376-55-4032  
7. 397-80-1857  
8. 723-43-4654  
9. 444-42-7451  
10. 864-47-3748  
11. 567-10-2981  
12. 999-74-1038  
13. 134-14-2948  
14. 321-52-3135  
15. 192-75-1145  
16. 943-85-2301  
17. 650-95-5849  
18. 994-82-1167  
19. 197-58-3670  
20. 110-39-9000  
21. 684-69-6437  
22. 682-29-8899  
23. 553-50-5063  
24. 265-96-6659  
25. 866-59-4503  
26. 464-48-9662  
27. 883-88-3558  
28. 303-93-4824  
29. 842-49-4881  
30. 843-84-9943  
31. 568-21-6088  
32. 580-68-6872  
33. 186-49-2470  
34. 945-61-4185  
35. 143-97-7442  
36. 116-16-4945  
37. 245-36-6379  
38. 560-93-4964  
39. 199-96-7832  
40. 893-64-4584  
41. 707-64-7802  
42. 524-82-9339  
43. 233-50-9811  
44. 772-39-1340  
45. 661-89-2996  
46. 720-71-7557  
47. 804-42-6578  
48. 540-72-8992  
49. 868-85-3408  
50. 331-15-9845  
51. 725-93-5169  
52. 486-34-9937  
53. 350-65-2468  
54. 337-17-2566  
55. 128-91-2778  
56. 266-54-7840  
57. 762-93-5888  
58. 748-38-7784  
59. 674-73-2624

1.



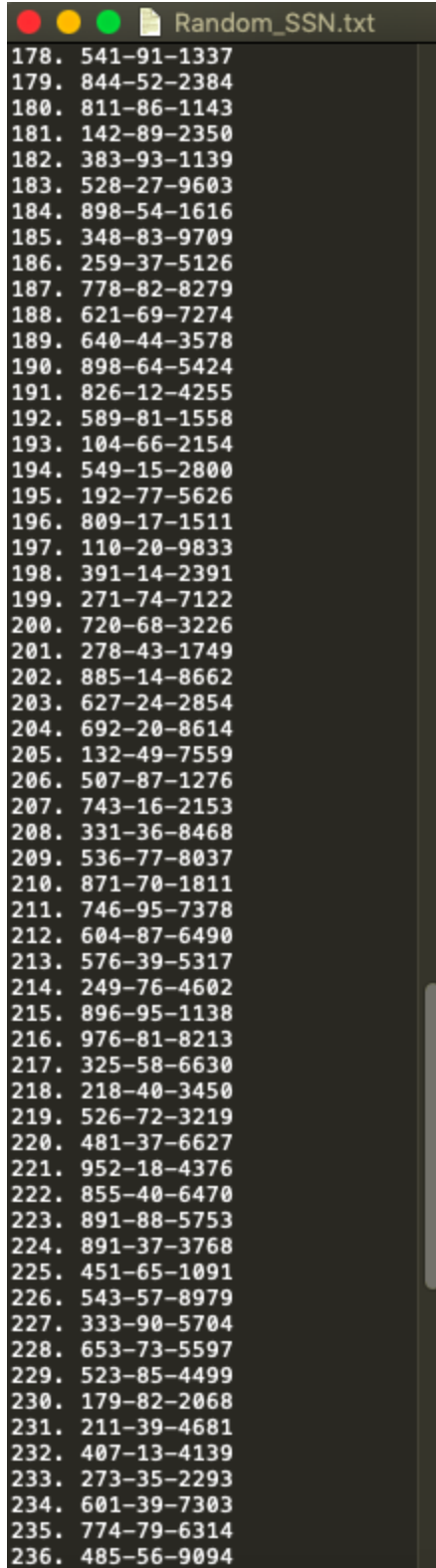
60. 466-39-4731  
61. 719-89-1492  
62. 835-28-1999  
63. 823-68-5195  
64. 991-44-4482  
65. 754-37-9616  
66. 144-19-6763  
67. 753-17-8574  
68. 966-62-8867  
69. 369-94-8971  
70. 958-99-8808  
71. 458-86-1651  
72. 347-10-6341  
73. 105-52-6572  
74. 317-88-2438  
75. 428-68-6570  
76. 173-12-4139  
77. 501-82-4879  
78. 113-78-5065  
79. 542-20-7580  
80. 762-31-5395  
81. 789-95-4694  
82. 532-35-7048  
83. 800-66-1663  
84. 829-73-6669  
85. 830-37-1597  
86. 239-45-2341  
87. 268-39-8824  
88. 821-89-2527  
89. 268-78-5751  
90. 984-82-2895  
91. 539-93-9831  
92. 907-16-1708  
93. 522-97-7893  
94. 147-90-5321  
95. 335-27-4384  
96. 336-63-3851  
97. 436-57-5010  
98. 731-27-9921  
99. 490-53-1288  
100. 720-35-8829  
101. 858-52-3903  
102. 792-16-2954  
103. 126-96-6068  
104. 302-93-9379  
105. 765-73-5200  
106. 575-74-4811  
107. 325-14-6772  
108. 519-26-1961  
109. 278-89-1767  
110. 559-16-4674  
111. 544-85-7241  
112. 715-50-7615  
113. 620-72-3537  
114. 880-28-7285  
115. 567-73-4117  
116. 693-48-5016  
117. 576-80-7801  
118. 373-35-6492

2.



119. 725-25-7112  
120. 390-39-9950  
121. 799-69-2830  
122. 759-40-3501  
123. 107-47-5689  
124. 705-88-2068  
125. 932-23-1344  
126. 875-92-7999  
127. 490-23-8285  
128. 401-82-7947  
129. 943-41-9912  
130. 776-79-8731  
131. 305-69-4521  
132. 308-16-5345  
133. 812-49-2660  
134. 212-18-5574  
135. 800-11-6486  
136. 421-48-8891  
137. 413-40-4929  
138. 729-86-6734  
139. 832-92-6201  
140. 534-85-5478  
141. 572-51-7542  
142. 344-36-6339  
143. 127-74-2700  
144. 851-21-1186  
145. 613-44-2227  
146. 521-24-9489  
147. 596-44-6861  
148. 342-80-7768  
149. 546-72-9499  
150. 260-58-1647  
151. 160-62-7148  
152. 457-32-9481  
153. 636-90-1868  
154. 641-28-6212  
155. 166-52-4814  
156. 370-65-9439  
157. 500-44-8661  
158. 655-75-3169  
159. 296-54-6803  
160. 147-42-5602  
161. 837-64-2027  
162. 797-80-4818  
163. 578-72-9694  
164. 244-57-9096  
165. 680-52-7791  
166. 716-79-3839  
167. 200-59-5404  
168. 374-41-1671  
169. 367-97-1779  
170. 877-99-2248  
171. 794-18-6141  
172. 411-83-6898  
173. 501-26-7043  
174. 311-76-7230  
175. 225-40-4162  
176. 536-12-5997  
177. 133-17-7045

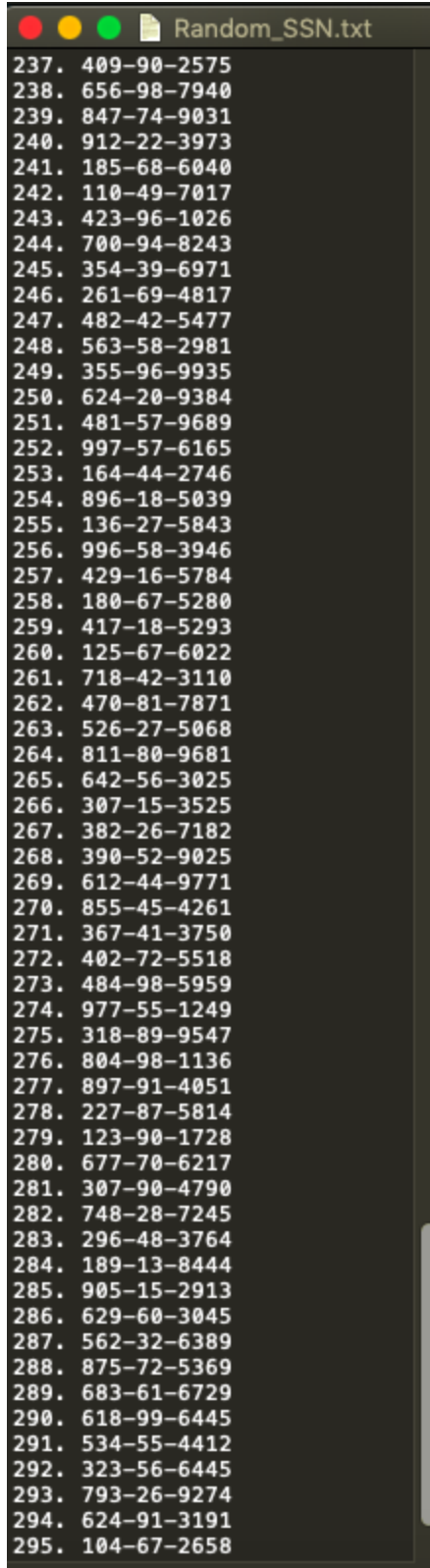
3.



Random\_SSN.txt

178. 541-91-1337  
179. 844-52-2384  
180. 811-86-1143  
181. 142-89-2350  
182. 383-93-1139  
183. 528-27-9603  
184. 898-54-1616  
185. 348-83-9709  
186. 259-37-5126  
187. 778-82-8279  
188. 621-69-7274  
189. 640-44-3578  
190. 898-64-5424  
191. 826-12-4255  
192. 589-81-1558  
193. 104-66-2154  
194. 549-15-2800  
195. 192-77-5626  
196. 809-17-1511  
197. 110-20-9833  
198. 391-14-2391  
199. 271-74-7122  
200. 720-68-3226  
201. 278-43-1749  
202. 885-14-8662  
203. 627-24-2854  
204. 692-20-8614  
205. 132-49-7559  
206. 507-87-1276  
207. 743-16-2153  
208. 331-36-8468  
209. 536-77-8037  
210. 871-70-1811  
211. 746-95-7378  
212. 604-87-6490  
213. 576-39-5317  
214. 249-76-4602  
215. 896-95-1138  
216. 976-81-8213  
217. 325-58-6630  
218. 218-40-3450  
219. 526-72-3219  
220. 481-37-6627  
221. 952-18-4376  
222. 855-40-6470  
223. 891-88-5753  
224. 891-37-3768  
225. 451-65-1091  
226. 543-57-8979  
227. 333-90-5704  
228. 653-73-5597  
229. 523-85-4499  
230. 179-82-2068  
231. 211-39-4681  
232. 407-13-4139  
233. 273-35-2293  
234. 601-39-7303  
235. 774-79-6314  
236. 485-56-9094

4.



237. 409-90-2575  
238. 656-98-7940  
239. 847-74-9031  
240. 912-22-3973  
241. 185-68-6040  
242. 110-49-7017  
243. 423-96-1026  
244. 700-94-8243  
245. 354-39-6971  
246. 261-69-4817  
247. 482-42-5477  
248. 563-58-2981  
249. 355-96-9935  
250. 624-20-9384  
251. 481-57-9689  
252. 997-57-6165  
253. 164-44-2746  
254. 896-18-5039  
255. 136-27-5843  
256. 996-58-3946  
257. 429-16-5784  
258. 180-67-5280  
259. 417-18-5293  
260. 125-67-6022  
261. 718-42-3110  
262. 470-81-7871  
263. 526-27-5068  
264. 811-80-9681  
265. 642-56-3025  
266. 307-15-3525  
267. 382-26-7182  
268. 390-52-9025  
269. 612-44-9771  
270. 855-45-4261  
271. 367-41-3750  
272. 402-72-5518  
273. 484-98-5959  
274. 977-55-1249  
275. 318-89-9547  
276. 804-98-1136  
277. 897-91-4051  
278. 227-87-5814  
279. 123-90-1728  
280. 677-70-6217  
281. 307-90-4790  
282. 748-28-7245  
283. 296-48-3764  
284. 189-13-8444  
285. 905-15-2913  
286. 629-60-3045  
287. 562-32-6389  
288. 875-72-5369  
289. 683-61-6729  
290. 618-99-6445  
291. 534-55-4412  
292. 323-56-6445  
293. 793-26-9274  
294. 624-91-3191  
295. 104-67-2658

5.

```
296. 988-87-9670
297. 840-32-9684
298. 484-19-7954
299. 711-24-9037
300. 266-31-1706
```

6.

b. Use Quick Sort on the 300 SSN and obtain the area count:

i. Input:

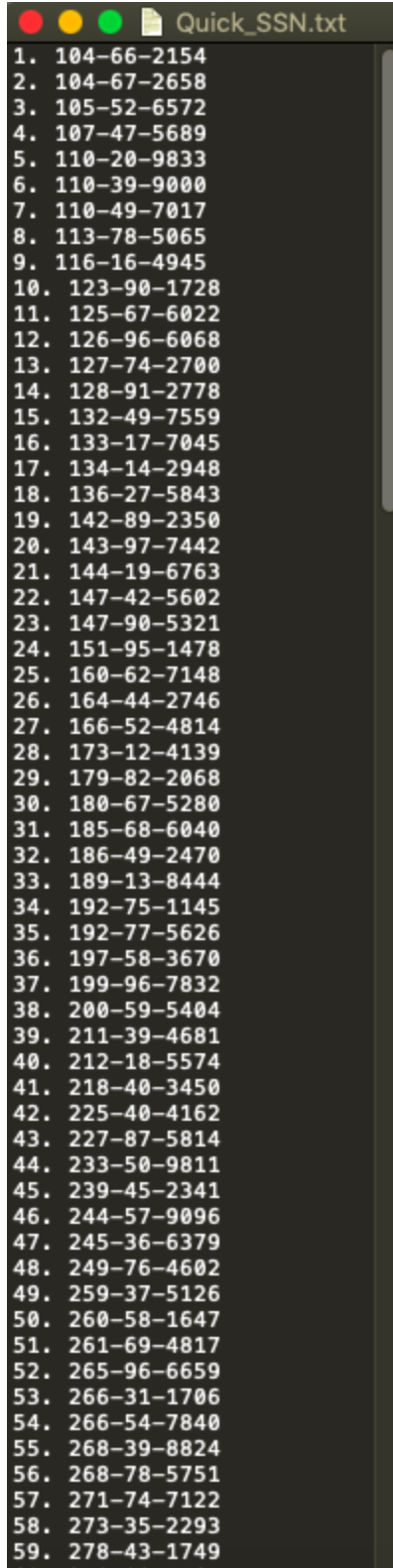
---

```
Please enter '2' to use Quick sort, find the population for each region, and write them into a text file.
Please enter '3' to use Radix sort, find the population for each region, and write them into a text file.
Please enter '4' to use Bucket sort, find the population for each region, and write them into a text file.
Please enter '5' if all options have been completed.
```

1. <sup>2</sup> The process has been completed. Please search your device with the label 'Quick\_SSN.txt' to see the file.

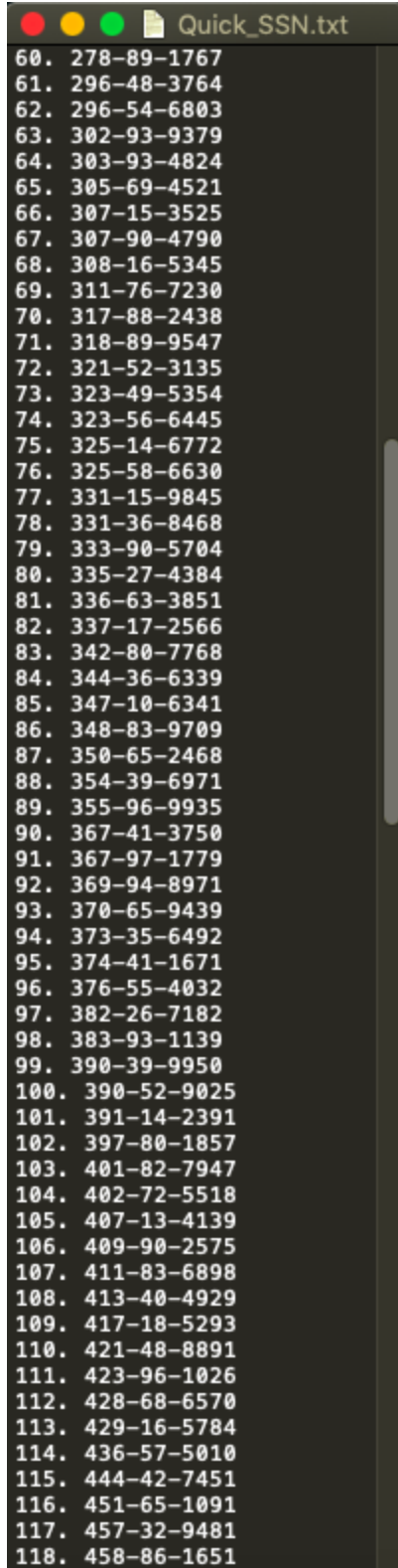
ii. Output:





1. 104-66-2154  
2. 104-67-2658  
3. 105-52-6572  
4. 107-47-5689  
5. 110-20-9833  
6. 110-39-9000  
7. 110-49-7017  
8. 113-78-5065  
9. 116-16-4945  
10. 123-90-1728  
11. 125-67-6022  
12. 126-96-6068  
13. 127-74-2700  
14. 128-91-2778  
15. 132-49-7559  
16. 133-17-7045  
17. 134-14-2948  
18. 136-27-5843  
19. 142-89-2350  
20. 143-97-7442  
21. 144-19-6763  
22. 147-42-5602  
23. 147-90-5321  
24. 151-95-1478  
25. 160-62-7148  
26. 164-44-2746  
27. 166-52-4814  
28. 173-12-4139  
29. 179-82-2068  
30. 180-67-5280  
31. 185-68-6040  
32. 186-49-2470  
33. 189-13-8444  
34. 192-75-1145  
35. 192-77-5626  
36. 197-58-3670  
37. 199-96-7832  
38. 200-59-5404  
39. 211-39-4681  
40. 212-18-5574  
41. 218-40-3450  
42. 225-40-4162  
43. 227-87-5814  
44. 233-50-9811  
45. 239-45-2341  
46. 244-57-9096  
47. 245-36-6379  
48. 249-76-4602  
49. 259-37-5126  
50. 260-58-1647  
51. 261-69-4817  
52. 265-96-6659  
53. 266-31-1706  
54. 266-54-7840  
55. 268-39-8824  
56. 268-78-5751  
57. 271-74-7122  
58. 273-35-2293  
59. 278-43-1749

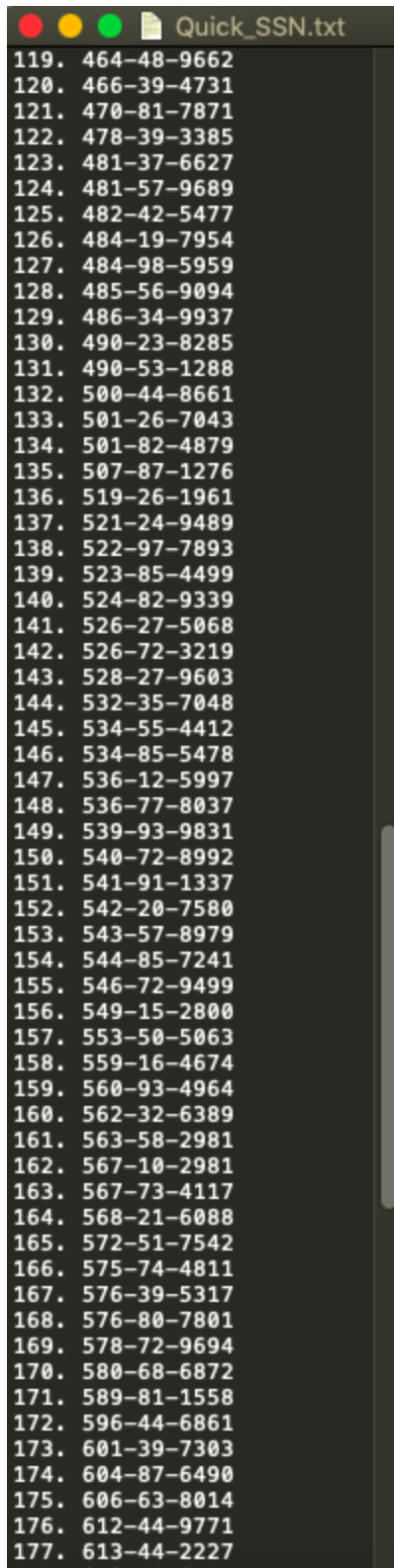
1.



Quick\_SSN.txt

60. 278-89-1767  
61. 296-48-3764  
62. 296-54-6803  
63. 302-93-9379  
64. 303-93-4824  
65. 305-69-4521  
66. 307-15-3525  
67. 307-90-4790  
68. 308-16-5345  
69. 311-76-7230  
70. 317-88-2438  
71. 318-89-9547  
72. 321-52-3135  
73. 323-49-5354  
74. 323-56-6445  
75. 325-14-6772  
76. 325-58-6630  
77. 331-15-9845  
78. 331-36-8468  
79. 333-90-5704  
80. 335-27-4384  
81. 336-63-3851  
82. 337-17-2566  
83. 342-80-7768  
84. 344-36-6339  
85. 347-10-6341  
86. 348-83-9709  
87. 350-65-2468  
88. 354-39-6971  
89. 355-96-9935  
90. 367-41-3750  
91. 367-97-1779  
92. 369-94-8971  
93. 370-65-9439  
94. 373-35-6492  
95. 374-41-1671  
96. 376-55-4032  
97. 382-26-7182  
98. 383-93-1139  
99. 390-39-9950  
100. 390-52-9025  
101. 391-14-2391  
102. 397-80-1857  
103. 401-82-7947  
104. 402-72-5518  
105. 407-13-4139  
106. 409-90-2575  
107. 411-83-6898  
108. 413-40-4929  
109. 417-18-5293  
110. 421-48-8891  
111. 423-96-1026  
112. 428-68-6570  
113. 429-16-5784  
114. 436-57-5010  
115. 444-42-7451  
116. 451-65-1091  
117. 457-32-9481  
118. 458-86-1651

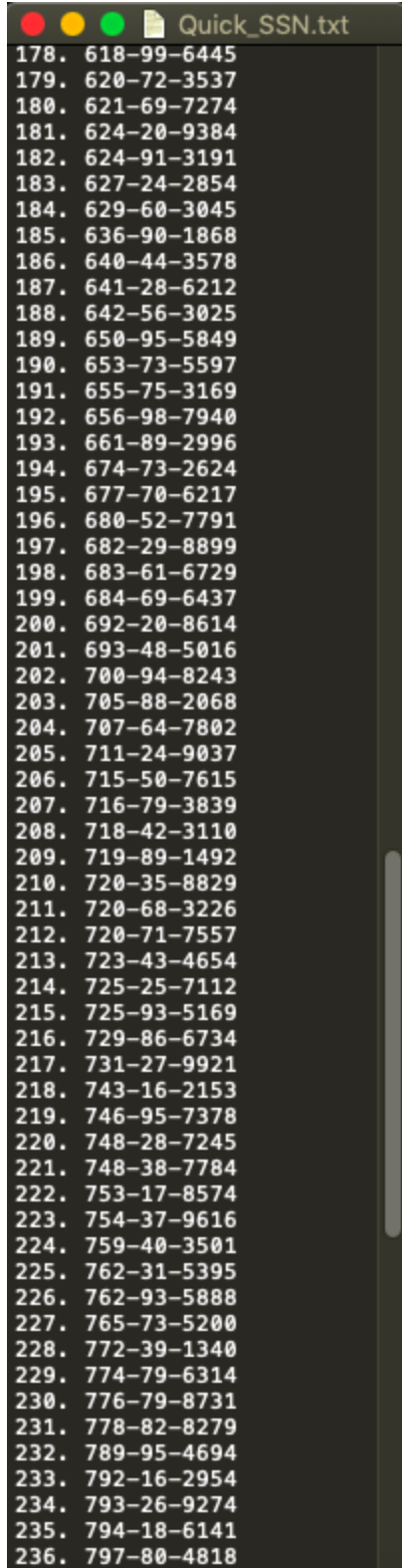
2.



Quick\_SSN.txt

119. 464-48-9662  
120. 466-39-4731  
121. 470-81-7871  
122. 478-39-3385  
123. 481-37-6627  
124. 481-57-9689  
125. 482-42-5477  
126. 484-19-7954  
127. 484-98-5959  
128. 485-56-9094  
129. 486-34-9937  
130. 490-23-8285  
131. 490-53-1288  
132. 500-44-8661  
133. 501-26-7043  
134. 501-82-4879  
135. 507-87-1276  
136. 519-26-1961  
137. 521-24-9489  
138. 522-97-7893  
139. 523-85-4499  
140. 524-82-9339  
141. 526-27-5068  
142. 526-72-3219  
143. 528-27-9603  
144. 532-35-7048  
145. 534-55-4412  
146. 534-85-5478  
147. 536-12-5997  
148. 536-77-8037  
149. 539-93-9831  
150. 540-72-8992  
151. 541-91-1337  
152. 542-20-7580  
153. 543-57-8979  
154. 544-85-7241  
155. 546-72-9499  
156. 549-15-2800  
157. 553-50-5063  
158. 559-16-4674  
159. 560-93-4964  
160. 562-32-6389  
161. 563-58-2981  
162. 567-10-2981  
163. 567-73-4117  
164. 568-21-6088  
165. 572-51-7542  
166. 575-74-4811  
167. 576-39-5317  
168. 576-80-7801  
169. 578-72-9694  
170. 580-68-6872  
171. 589-81-1558  
172. 596-44-6861  
173. 601-39-7303  
174. 604-87-6490  
175. 606-63-8014  
176. 612-44-9771  
177. 613-44-2227

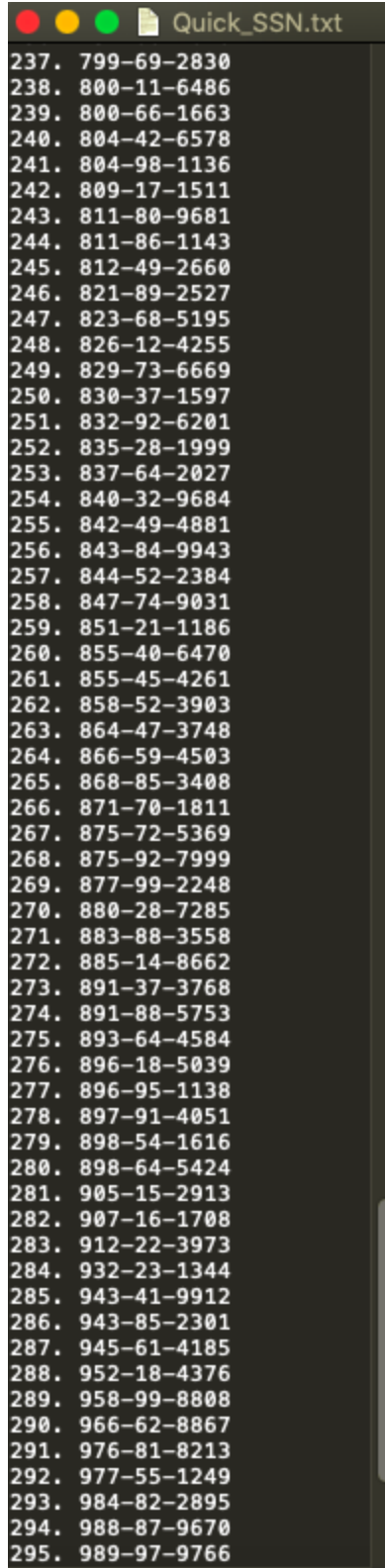
3.



Quick\_SSN.txt

178. 618-99-6445  
179. 620-72-3537  
180. 621-69-7274  
181. 624-20-9384  
182. 624-91-3191  
183. 627-24-2854  
184. 629-60-3045  
185. 636-90-1868  
186. 640-44-3578  
187. 641-28-6212  
188. 642-56-3025  
189. 650-95-5849  
190. 653-73-5597  
191. 655-75-3169  
192. 656-98-7940  
193. 661-89-2996  
194. 674-73-2624  
195. 677-70-6217  
196. 680-52-7791  
197. 682-29-8899  
198. 683-61-6729  
199. 684-69-6437  
200. 692-20-8614  
201. 693-48-5016  
202. 700-94-8243  
203. 705-88-2068  
204. 707-64-7802  
205. 711-24-9037  
206. 715-50-7615  
207. 716-79-3839  
208. 718-42-3110  
209. 719-89-1492  
210. 720-35-8829  
211. 720-68-3226  
212. 720-71-7557  
213. 723-43-4654  
214. 725-25-7112  
215. 725-93-5169  
216. 729-86-6734  
217. 731-27-9921  
218. 743-16-2153  
219. 746-95-7378  
220. 748-28-7245  
221. 748-38-7784  
222. 753-17-8574  
223. 754-37-9616  
224. 759-40-3501  
225. 762-31-5395  
226. 762-93-5888  
227. 765-73-5200  
228. 772-39-1340  
229. 774-79-6314  
230. 776-79-8731  
231. 778-82-8279  
232. 789-95-4694  
233. 792-16-2954  
234. 793-26-9274  
235. 794-18-6141  
236. 797-80-4818

4.



Quick\_SSN.txt

237. 799-69-2830  
238. 800-11-6486  
239. 800-66-1663  
240. 804-42-6578  
241. 804-98-1136  
242. 809-17-1511  
243. 811-80-9681  
244. 811-86-1143  
245. 812-49-2660  
246. 821-89-2527  
247. 823-68-5195  
248. 826-12-4255  
249. 829-73-6669  
250. 830-37-1597  
251. 832-92-6201  
252. 835-28-1999  
253. 837-64-2027  
254. 840-32-9684  
255. 842-49-4881  
256. 843-84-9943  
257. 844-52-2384  
258. 847-74-9031  
259. 851-21-1186  
260. 855-40-6470  
261. 855-45-4261  
262. 858-52-3903  
263. 864-47-3748  
264. 866-59-4503  
265. 868-85-3408  
266. 871-70-1811  
267. 875-72-5369  
268. 875-92-7999  
269. 877-99-2248  
270. 880-28-7285  
271. 883-88-3558  
272. 885-14-8662  
273. 891-37-3768  
274. 891-88-5753  
275. 893-64-4584  
276. 896-18-5039  
277. 896-95-1138  
278. 897-91-4051  
279. 898-54-1616  
280. 898-64-5424  
281. 905-15-2913  
282. 907-16-1708  
283. 912-22-3973  
284. 932-23-1344  
285. 943-41-9912  
286. 943-85-2301  
287. 945-61-4185  
288. 952-18-4376  
289. 958-99-8808  
290. 966-62-8867  
291. 976-81-8213  
292. 977-55-1249  
293. 984-82-2895  
294. 988-87-9670  
295. 989-97-9766

5.

```
296. 991-44-4482
297. 994-82-1167
298. 996-58-3946
299. 997-57-6165
300. 999-74-1038

Northeast Coast States: 37 people
South Coast States: 65 people
Middle States: 70 people
Northwest Coast States: 65 people
West Coast States: 63 people
```

6.

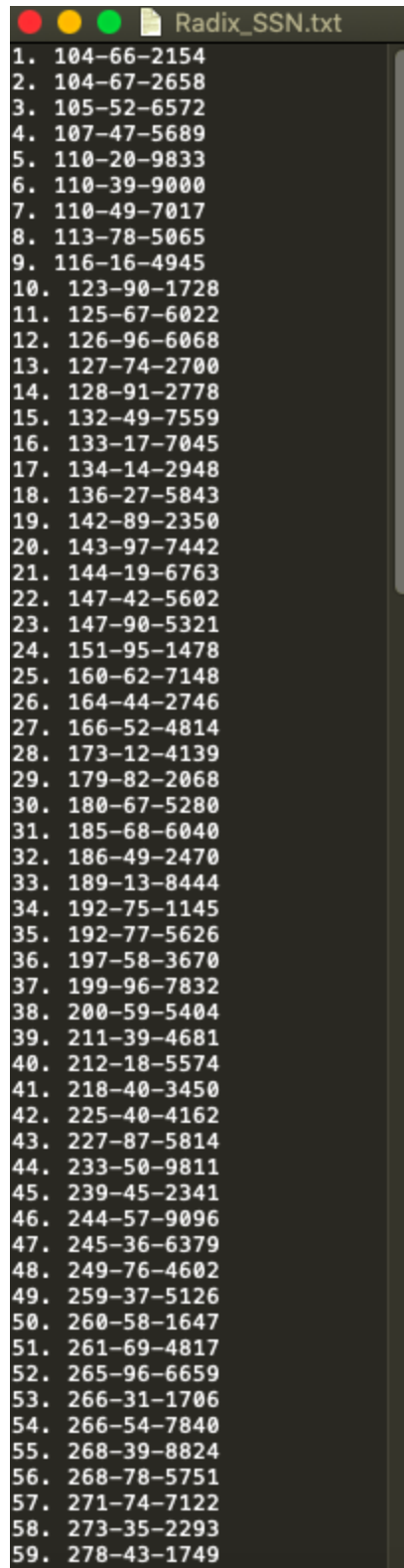
c. Use Radix Sort on the 300 SSN and obtain the area count:

i. Input:

```
Please enter '2' to use Quick sort, find the population for each region, and write them into a text file.
Please enter '3' to use Radix sort, find the population for each region, and write them into a text file.
Please enter '4' to use Bucket sort, find the population for each region, and write them into a text file.
Please enter '5' if all options have been completed.
```

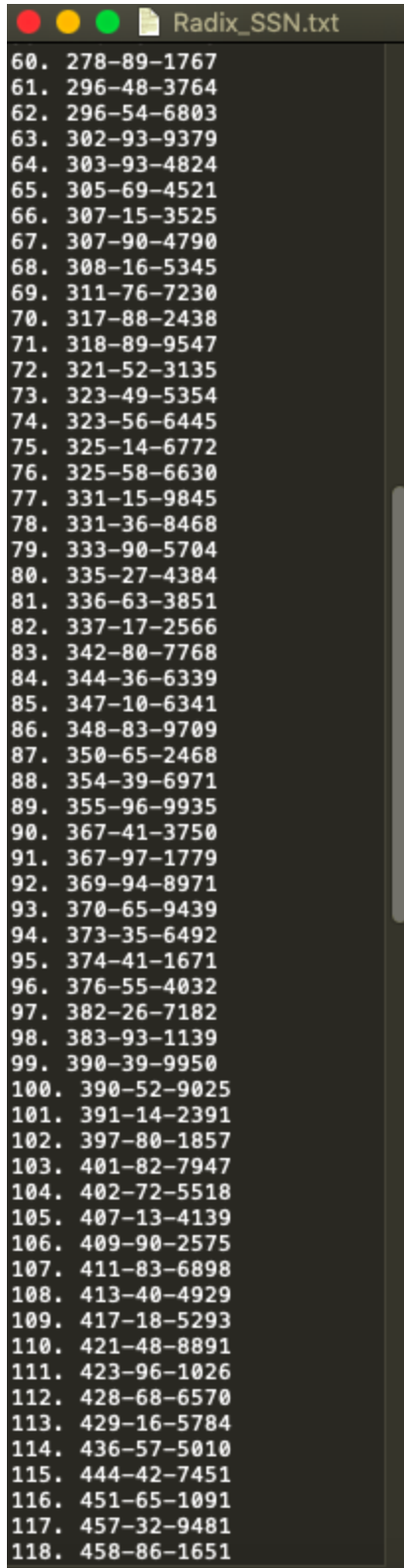
1. 3  
The process has been completed. Please search your device with the label 'Radix\_SSN.txt' to see the file.

ii. Output:



```
1. 104-66-2154
2. 104-67-2658
3. 105-52-6572
4. 107-47-5689
5. 110-20-9833
6. 110-39-9000
7. 110-49-7017
8. 113-78-5065
9. 116-16-4945
10. 123-90-1728
11. 125-67-6022
12. 126-96-6068
13. 127-74-2700
14. 128-91-2778
15. 132-49-7559
16. 133-17-7045
17. 134-14-2948
18. 136-27-5843
19. 142-89-2350
20. 143-97-7442
21. 144-19-6763
22. 147-42-5602
23. 147-90-5321
24. 151-95-1478
25. 160-62-7148
26. 164-44-2746
27. 166-52-4814
28. 173-12-4139
29. 179-82-2068
30. 180-67-5280
31. 185-68-6040
32. 186-49-2470
33. 189-13-8444
34. 192-75-1145
35. 192-77-5626
36. 197-58-3670
37. 199-96-7832
38. 200-59-5404
39. 211-39-4681
40. 212-18-5574
41. 218-40-3450
42. 225-40-4162
43. 227-87-5814
44. 233-50-9811
45. 239-45-2341
46. 244-57-9096
47. 245-36-6379
48. 249-76-4602
49. 259-37-5126
50. 260-58-1647
51. 261-69-4817
52. 265-96-6659
53. 266-31-1706
54. 266-54-7840
55. 268-39-8824
56. 268-78-5751
57. 271-74-7122
58. 273-35-2293
59. 278-43-1749
```

1.

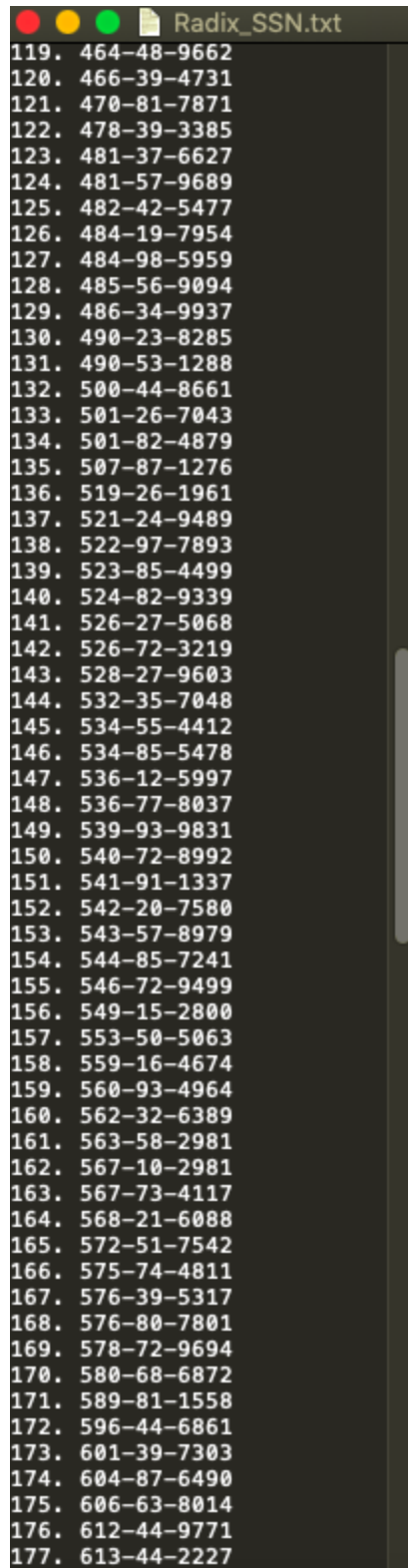


Radix\_SSN.txt

60. 278-89-1767  
61. 296-48-3764  
62. 296-54-6803  
63. 302-93-9379  
64. 303-93-4824  
65. 305-69-4521  
66. 307-15-3525  
67. 307-90-4790  
68. 308-16-5345  
69. 311-76-7230  
70. 317-88-2438  
71. 318-89-9547  
72. 321-52-3135  
73. 323-49-5354  
74. 323-56-6445  
75. 325-14-6772  
76. 325-58-6630  
77. 331-15-9845  
78. 331-36-8468  
79. 333-90-5704  
80. 335-27-4384  
81. 336-63-3851  
82. 337-17-2566  
83. 342-80-7768  
84. 344-36-6339  
85. 347-10-6341  
86. 348-83-9709  
87. 350-65-2468  
88. 354-39-6971  
89. 355-96-9935  
90. 367-41-3750  
91. 367-97-1779  
92. 369-94-8971  
93. 370-65-9439  
94. 373-35-6492  
95. 374-41-1671  
96. 376-55-4032  
97. 382-26-7182  
98. 383-93-1139  
99. 390-39-9950  
100. 390-52-9025  
101. 391-14-2391  
102. 397-80-1857  
103. 401-82-7947  
104. 402-72-5518  
105. 407-13-4139  
106. 409-90-2575  
107. 411-83-6898  
108. 413-40-4929  
109. 417-18-5293  
110. 421-48-8891  
111. 423-96-1026  
112. 428-68-6570  
113. 429-16-5784  
114. 436-57-5010  
115. 444-42-7451  
116. 451-65-1091  
117. 457-32-9481  
118. 458-86-1651

2.

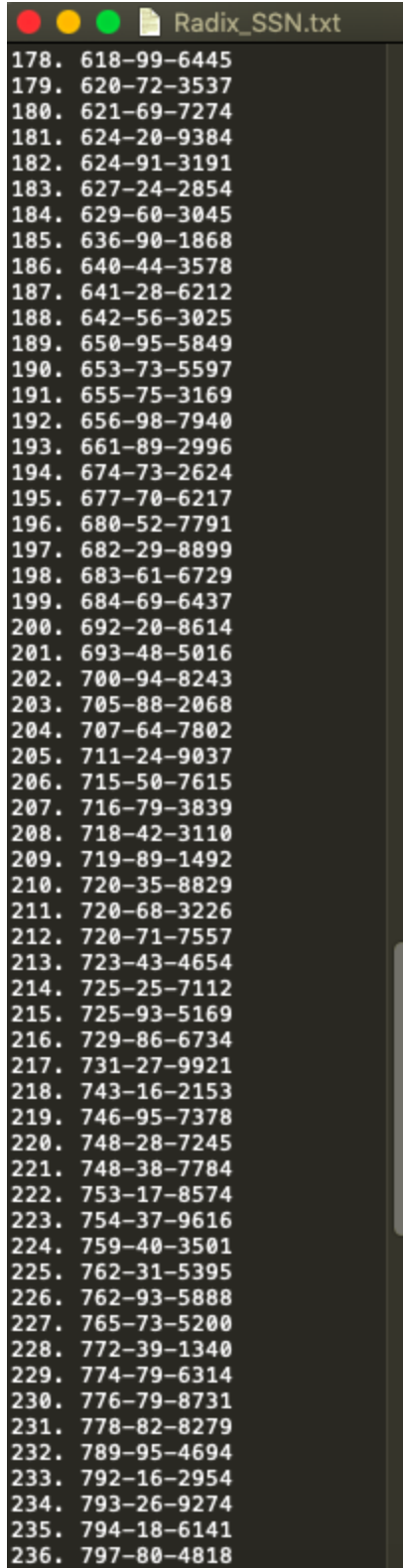




Radix\_SSN.txt

119. 464-48-9662  
120. 466-39-4731  
121. 470-81-7871  
122. 478-39-3385  
123. 481-37-6627  
124. 481-57-9689  
125. 482-42-5477  
126. 484-19-7954  
127. 484-98-5959  
128. 485-56-9094  
129. 486-34-9937  
130. 490-23-8285  
131. 490-53-1288  
132. 500-44-8661  
133. 501-26-7043  
134. 501-82-4879  
135. 507-87-1276  
136. 519-26-1961  
137. 521-24-9489  
138. 522-97-7893  
139. 523-85-4499  
140. 524-82-9339  
141. 526-27-5068  
142. 526-72-3219  
143. 528-27-9603  
144. 532-35-7048  
145. 534-55-4412  
146. 534-85-5478  
147. 536-12-5997  
148. 536-77-8037  
149. 539-93-9831  
150. 540-72-8992  
151. 541-91-1337  
152. 542-20-7580  
153. 543-57-8979  
154. 544-85-7241  
155. 546-72-9499  
156. 549-15-2800  
157. 553-50-5063  
158. 559-16-4674  
159. 560-93-4964  
160. 562-32-6389  
161. 563-58-2981  
162. 567-10-2981  
163. 567-73-4117  
164. 568-21-6088  
165. 572-51-7542  
166. 575-74-4811  
167. 576-39-5317  
168. 576-80-7801  
169. 578-72-9694  
170. 580-68-6872  
171. 589-81-1558  
172. 596-44-6861  
173. 601-39-7303  
174. 604-87-6490  
175. 606-63-8014  
176. 612-44-9771  
177. 613-44-2227

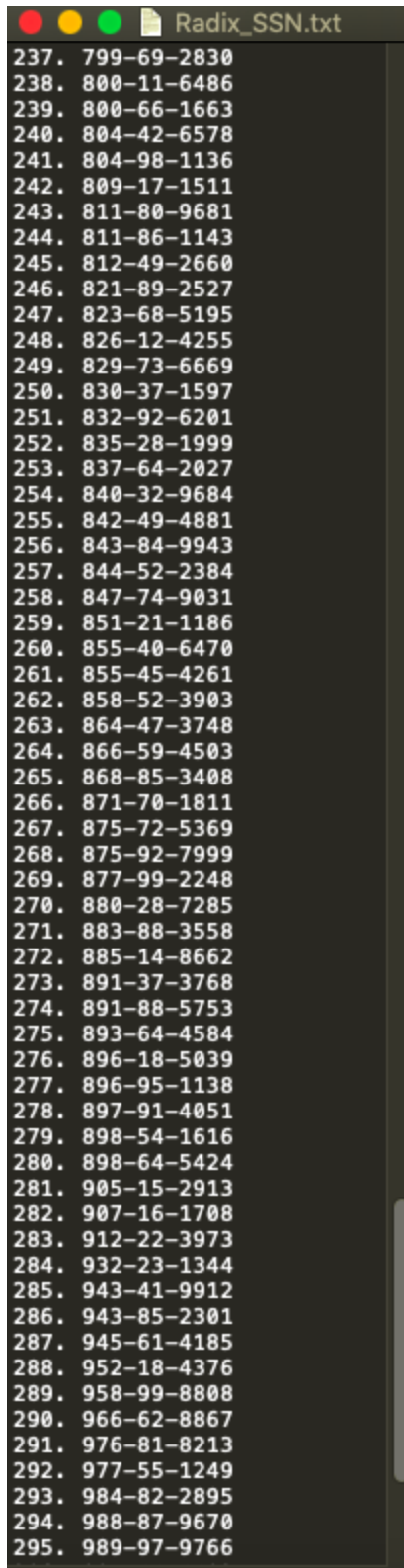
3.



Radix\_SSN.txt

178. 618-99-6445  
179. 620-72-3537  
180. 621-69-7274  
181. 624-20-9384  
182. 624-91-3191  
183. 627-24-2854  
184. 629-60-3045  
185. 636-90-1868  
186. 640-44-3578  
187. 641-28-6212  
188. 642-56-3025  
189. 650-95-5849  
190. 653-73-5597  
191. 655-75-3169  
192. 656-98-7940  
193. 661-89-2996  
194. 674-73-2624  
195. 677-70-6217  
196. 680-52-7791  
197. 682-29-8899  
198. 683-61-6729  
199. 684-69-6437  
200. 692-20-8614  
201. 693-48-5016  
202. 700-94-8243  
203. 705-88-2068  
204. 707-64-7802  
205. 711-24-9037  
206. 715-50-7615  
207. 716-79-3839  
208. 718-42-3110  
209. 719-89-1492  
210. 720-35-8829  
211. 720-68-3226  
212. 720-71-7557  
213. 723-43-4654  
214. 725-25-7112  
215. 725-93-5169  
216. 729-86-6734  
217. 731-27-9921  
218. 743-16-2153  
219. 746-95-7378  
220. 748-28-7245  
221. 748-38-7784  
222. 753-17-8574  
223. 754-37-9616  
224. 759-40-3501  
225. 762-31-5395  
226. 762-93-5888  
227. 765-73-5200  
228. 772-39-1340  
229. 774-79-6314  
230. 776-79-8731  
231. 778-82-8279  
232. 789-95-4694  
233. 792-16-2954  
234. 793-26-9274  
235. 794-18-6141  
236. 797-80-4818

4.



Radix\_SSN.txt

237. 799-69-2830  
238. 800-11-6486  
239. 800-66-1663  
240. 804-42-6578  
241. 804-98-1136  
242. 809-17-1511  
243. 811-80-9681  
244. 811-86-1143  
245. 812-49-2660  
246. 821-89-2527  
247. 823-68-5195  
248. 826-12-4255  
249. 829-73-6669  
250. 830-37-1597  
251. 832-92-6201  
252. 835-28-1999  
253. 837-64-2027  
254. 840-32-9684  
255. 842-49-4881  
256. 843-84-9943  
257. 844-52-2384  
258. 847-74-9031  
259. 851-21-1186  
260. 855-40-6470  
261. 855-45-4261  
262. 858-52-3903  
263. 864-47-3748  
264. 866-59-4503  
265. 868-85-3408  
266. 871-70-1811  
267. 875-72-5369  
268. 875-92-7999  
269. 877-99-2248  
270. 880-28-7285  
271. 883-88-3558  
272. 885-14-8662  
273. 891-37-3768  
274. 891-88-5753  
275. 893-64-4584  
276. 896-18-5039  
277. 896-95-1138  
278. 897-91-4051  
279. 898-54-1616  
280. 898-64-5424  
281. 905-15-2913  
282. 907-16-1708  
283. 912-22-3973  
284. 932-23-1344  
285. 943-41-9912  
286. 943-85-2301  
287. 945-61-4185  
288. 952-18-4376  
289. 958-99-8808  
290. 966-62-8867  
291. 976-81-8213  
292. 977-55-1249  
293. 984-82-2895  
294. 988-87-9670  
295. 989-97-9766

5.

```
296. 991-44-4482
297. 994-82-1167
298. 996-58-3946
299. 997-57-6165
300. 999-74-1038

Northeast Coast States: 37 people
South Coast States: 65 people
Middle States: 70 people
Northwest Coast States: 65 people
West Coast States: 63 people
```

6.

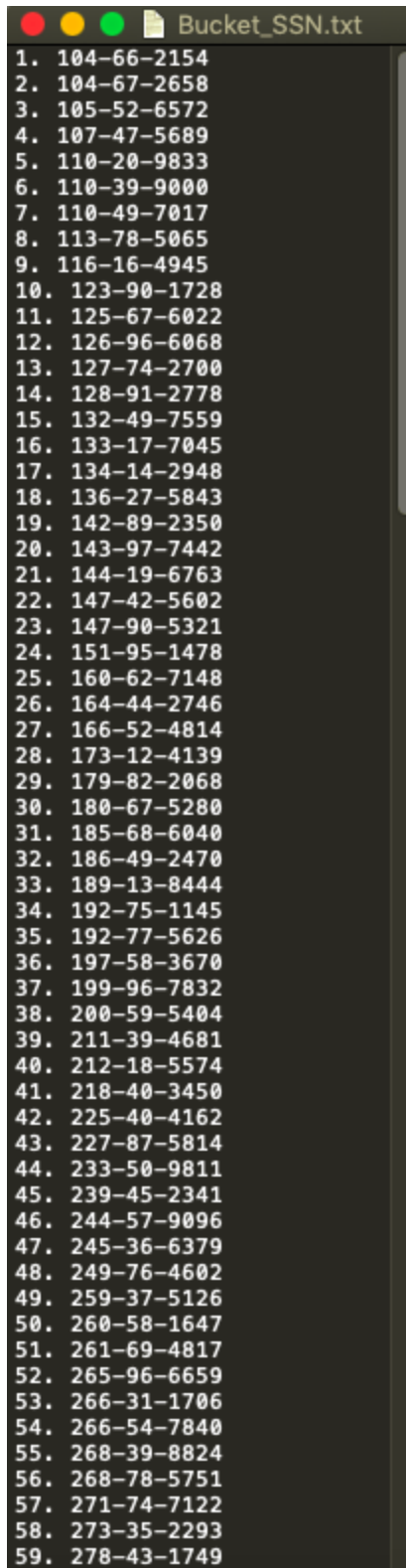
d. Use Bucket Sort on the 300 SSN and obtain the area count:

i. Input:

```
Please enter '2' to use Quick sort, find the population for each region, and write them into a text file.
Please enter '3' to use Radix sort, find the population for each region, and write them into a text file.
Please enter '4' to use Bucket sort, find the population for each region, and write them into a text file.
Please enter '5' if all options have been completed.
```

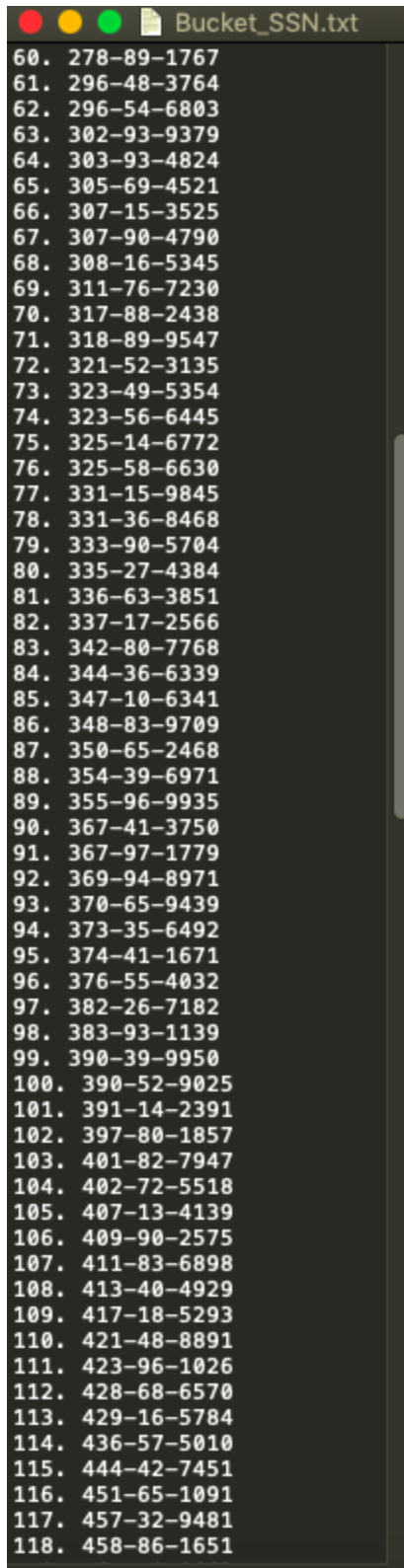
1. The process has been completed. Please search your device with the label 'Bucket\_SSN.txt' to see the file.

ii. Output:



1. 104-66-2154  
2. 104-67-2658  
3. 105-52-6572  
4. 107-47-5689  
5. 110-20-9833  
6. 110-39-9000  
7. 110-49-7017  
8. 113-78-5065  
9. 116-16-4945  
10. 123-90-1728  
11. 125-67-6022  
12. 126-96-6068  
13. 127-74-2700  
14. 128-91-2778  
15. 132-49-7559  
16. 133-17-7045  
17. 134-14-2948  
18. 136-27-5843  
19. 142-89-2350  
20. 143-97-7442  
21. 144-19-6763  
22. 147-42-5602  
23. 147-90-5321  
24. 151-95-1478  
25. 160-62-7148  
26. 164-44-2746  
27. 166-52-4814  
28. 173-12-4139  
29. 179-82-2068  
30. 180-67-5280  
31. 185-68-6040  
32. 186-49-2470  
33. 189-13-8444  
34. 192-75-1145  
35. 192-77-5626  
36. 197-58-3670  
37. 199-96-7832  
38. 200-59-5404  
39. 211-39-4681  
40. 212-18-5574  
41. 218-40-3450  
42. 225-40-4162  
43. 227-87-5814  
44. 233-50-9811  
45. 239-45-2341  
46. 244-57-9096  
47. 245-36-6379  
48. 249-76-4602  
49. 259-37-5126  
50. 260-58-1647  
51. 261-69-4817  
52. 265-96-6659  
53. 266-31-1706  
54. 266-54-7840  
55. 268-39-8824  
56. 268-78-5751  
57. 271-74-7122  
58. 273-35-2293  
59. 278-43-1749

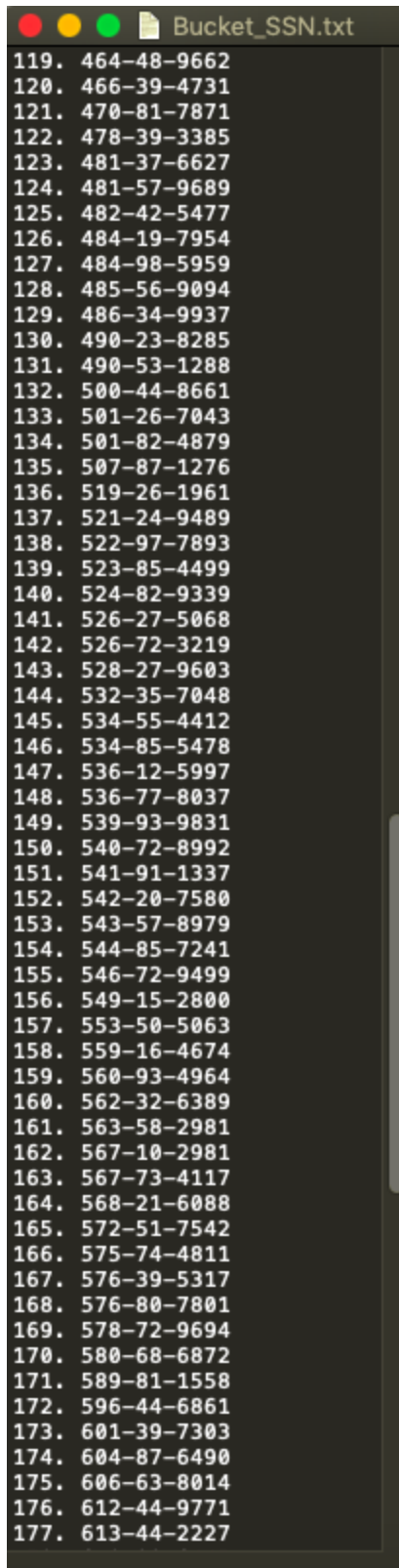
1.



Bucket\_SSN.txt

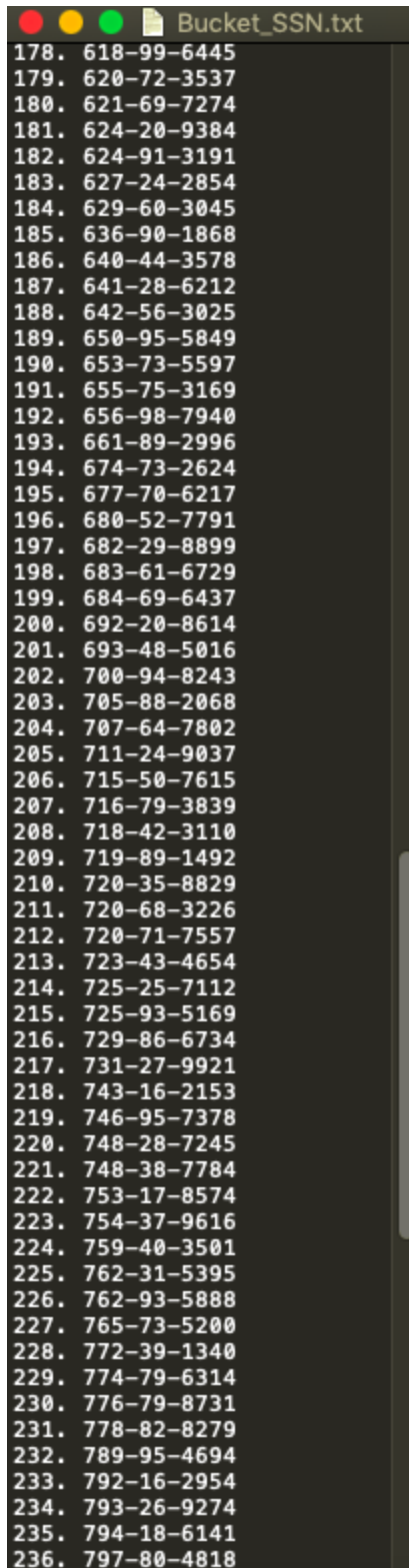
60. 278-89-1767  
61. 296-48-3764  
62. 296-54-6803  
63. 302-93-9379  
64. 303-93-4824  
65. 305-69-4521  
66. 307-15-3525  
67. 307-90-4790  
68. 308-16-5345  
69. 311-76-7230  
70. 317-88-2438  
71. 318-89-9547  
72. 321-52-3135  
73. 323-49-5354  
74. 323-56-6445  
75. 325-14-6772  
76. 325-58-6630  
77. 331-15-9845  
78. 331-36-8468  
79. 333-90-5704  
80. 335-27-4384  
81. 336-63-3851  
82. 337-17-2566  
83. 342-80-7768  
84. 344-36-6339  
85. 347-10-6341  
86. 348-83-9709  
87. 350-65-2468  
88. 354-39-6971  
89. 355-96-9935  
90. 367-41-3750  
91. 367-97-1779  
92. 369-94-8971  
93. 370-65-9439  
94. 373-35-6492  
95. 374-41-1671  
96. 376-55-4032  
97. 382-26-7182  
98. 383-93-1139  
99. 390-39-9950  
100. 390-52-9025  
101. 391-14-2391  
102. 397-80-1857  
103. 401-82-7947  
104. 402-72-5518  
105. 407-13-4139  
106. 409-90-2575  
107. 411-83-6898  
108. 413-40-4929  
109. 417-18-5293  
110. 421-48-8891  
111. 423-96-1026  
112. 428-68-6570  
113. 429-16-5784  
114. 436-57-5010  
115. 444-42-7451  
116. 451-65-1091  
117. 457-32-9481  
118. 458-86-1651

2.



119. 464-48-9662  
120. 466-39-4731  
121. 470-81-7871  
122. 478-39-3385  
123. 481-37-6627  
124. 481-57-9689  
125. 482-42-5477  
126. 484-19-7954  
127. 484-98-5959  
128. 485-56-9094  
129. 486-34-9937  
130. 490-23-8285  
131. 490-53-1288  
132. 500-44-8661  
133. 501-26-7043  
134. 501-82-4879  
135. 507-87-1276  
136. 519-26-1961  
137. 521-24-9489  
138. 522-97-7893  
139. 523-85-4499  
140. 524-82-9339  
141. 526-27-5068  
142. 526-72-3219  
143. 528-27-9603  
144. 532-35-7048  
145. 534-55-4412  
146. 534-85-5478  
147. 536-12-5997  
148. 536-77-8037  
149. 539-93-9831  
150. 540-72-8992  
151. 541-91-1337  
152. 542-20-7580  
153. 543-57-8979  
154. 544-85-7241  
155. 546-72-9499  
156. 549-15-2800  
157. 553-50-5063  
158. 559-16-4674  
159. 560-93-4964  
160. 562-32-6389  
161. 563-58-2981  
162. 567-10-2981  
163. 567-73-4117  
164. 568-21-6088  
165. 572-51-7542  
166. 575-74-4811  
167. 576-39-5317  
168. 576-80-7801  
169. 578-72-9694  
170. 580-68-6872  
171. 589-81-1558  
172. 596-44-6861  
173. 601-39-7303  
174. 604-87-6490  
175. 606-63-8014  
176. 612-44-9771  
177. 613-44-2227

3.



Bucket\_SSN.txt

178. 618-99-6445  
179. 620-72-3537  
180. 621-69-7274  
181. 624-20-9384  
182. 624-91-3191  
183. 627-24-2854  
184. 629-60-3045  
185. 636-90-1868  
186. 640-44-3578  
187. 641-28-6212  
188. 642-56-3025  
189. 650-95-5849  
190. 653-73-5597  
191. 655-75-3169  
192. 656-98-7940  
193. 661-89-2996  
194. 674-73-2624  
195. 677-70-6217  
196. 680-52-7791  
197. 682-29-8899  
198. 683-61-6729  
199. 684-69-6437  
200. 692-20-8614  
201. 693-48-5016  
202. 700-94-8243  
203. 705-88-2068  
204. 707-64-7802  
205. 711-24-9037  
206. 715-50-7615  
207. 716-79-3839  
208. 718-42-3110  
209. 719-89-1492  
210. 720-35-8829  
211. 720-68-3226  
212. 720-71-7557  
213. 723-43-4654  
214. 725-25-7112  
215. 725-93-5169  
216. 729-86-6734  
217. 731-27-9921  
218. 743-16-2153  
219. 746-95-7378  
220. 748-28-7245  
221. 748-38-7784  
222. 753-17-8574  
223. 754-37-9616  
224. 759-40-3501  
225. 762-31-5395  
226. 762-93-5888  
227. 765-73-5200  
228. 772-39-1340  
229. 774-79-6314  
230. 776-79-8731  
231. 778-82-8279  
232. 789-95-4694  
233. 792-16-2954  
234. 793-26-9274  
235. 794-18-6141  
236. 797-80-4818

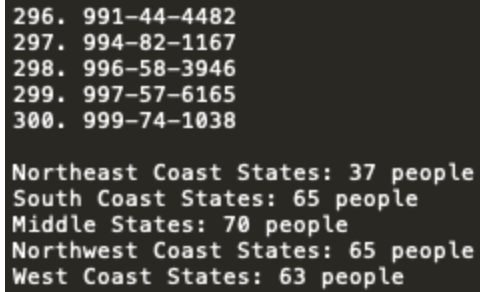
4.



Bucket\_SSN.txt

237. 799-69-2830  
238. 800-11-6486  
239. 800-66-1663  
240. 804-42-6578  
241. 804-98-1136  
242. 809-17-1511  
243. 811-80-9681  
244. 811-86-1143  
245. 812-49-2660  
246. 821-89-2527  
247. 823-68-5195  
248. 826-12-4255  
249. 829-73-6669  
250. 830-37-1597  
251. 832-92-6201  
252. 835-28-1999  
253. 837-64-2027  
254. 840-32-9684  
255. 842-49-4881  
256. 843-84-9943  
257. 844-52-2384  
258. 847-74-9031  
259. 851-21-1186  
260. 855-40-6470  
261. 855-45-4261  
262. 858-52-3903  
263. 864-47-3748  
264. 866-59-4503  
265. 868-85-3408  
266. 871-70-1811  
267. 875-72-5369  
268. 875-92-7999  
269. 877-99-2248  
270. 880-28-7285  
271. 883-88-3558  
272. 885-14-8662  
273. 891-37-3768  
274. 891-88-5753  
275. 893-64-4584  
276. 896-18-5039  
277. 896-95-1138  
278. 897-91-4051  
279. 898-54-1616  
280. 898-64-5424  
281. 905-15-2913  
282. 907-16-1708  
283. 912-22-3973  
284. 932-23-1344  
285. 943-41-9912  
286. 943-85-2301  
287. 945-61-4185  
288. 952-18-4376  
289. 958-99-8808  
290. 966-62-8867  
291. 976-81-8213  
292. 977-55-1249  
293. 984-82-2895  
294. 988-87-9670  
295. 989-97-9766

5.



```
296. 991-44-4482
297. 994-82-1167
298. 996-58-3946
299. 997-57-6165
300. 999-74-1038

Northeast Coast States: 37 people
South Coast States: 65 people
Middle States: 70 people
Northwest Coast States: 65 people
West Coast States: 63 people
```

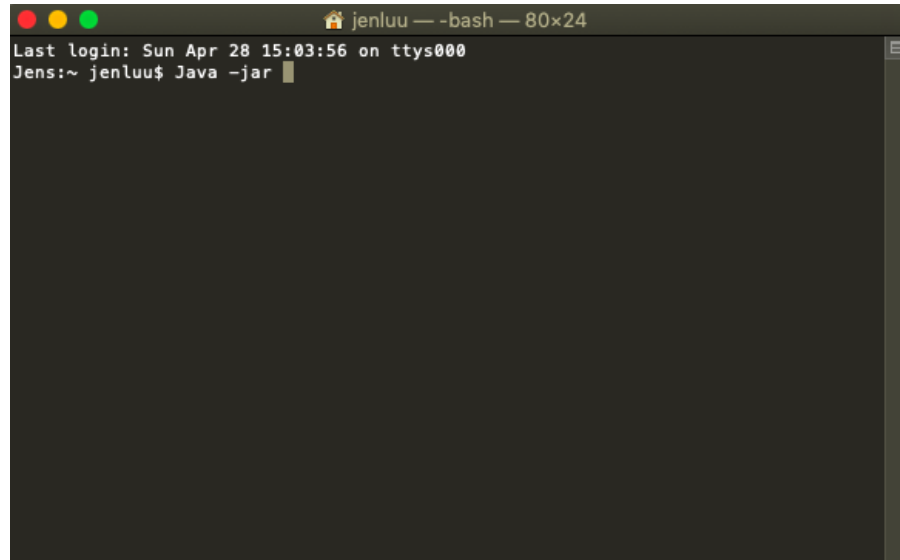
6.

#### D. How to Run the Program

- a. Locate the zipped file labeled “PA2-Section 7-Luu” and double click to unzip.

The following 11 files should appear:

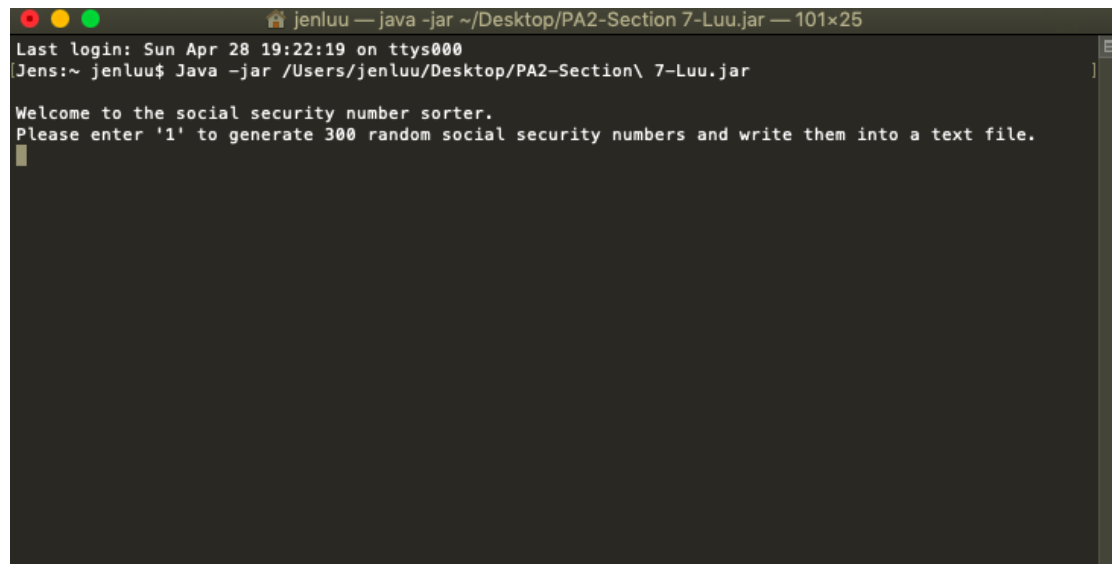
- i. Number.java
  - ii. Quicksort.java
  - iii. RadixSort.java
  - iv. BucketSort.java
  - v. Sorter.java
  - vi. Random\_SSN.txt
  - vii. Quick\_SSN.txt
  - viii. Radix\_SSN.txt
  - ix. Bucket\_SSN.txt
  - x. PA2-Section 7-Luu Report.docx
  - xi. PA2-Section 7-Luu.jar
- b. Open the Terminal. Type in “Java -jar .” It is important to leave a space after the “jar.”



```
jenluu — -bash — 80x24
Last login: Sun Apr 28 15:03:56 on ttys000
Jens:~ jenluu$ Java -jar
```

i.

- c. Drag the jar file “PA2-Section 7-Luu.jar” into the Terminal and press “Enter.” The program will begin to prompt the user.



```
jenluu — java -jar ~/Desktop/PA2-Section 7-Luu.jar — 101x25
Last login: Sun Apr 28 19:22:19 on ttys000
Jens:~ jenluu$ Java -jar /Users/jenluu/Desktop/PA2-Section\ 7-Luu.jar

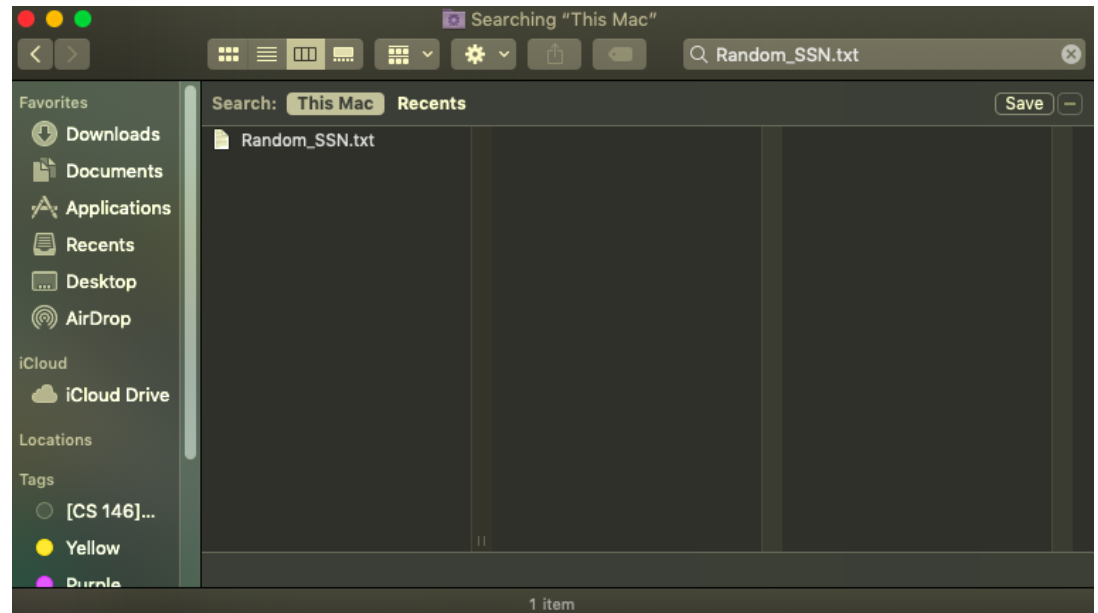
Welcome to the social security number sorter.
Please enter '1' to generate 300 random social security numbers and write them into a text file.

```

i.

- d. To search for the text files, open the Finder on the Mac and type in the desired file. Make sure to click “This Mac” to ensure a comprehensive search.

i.



## E. Problems Encountered

### a. Defining the digit for Radix sort

- i. It is one thing to go through an algorithm by hand, and it is an entirely different thing to code it. One of my major mistakes was depending on the pseudocode in the textbook to have all of the steps to code Radix sort. Originally, I thought that if I were to code each line according to the textbook then I could move on. However, that was far from reality. While the textbook explains the concept of each step of Radix sort, the user must translate those concepts according to the requirements of the function. Therefore, I had a challenging time trying to write the code to sort digit by digit since there was no built-in method to obtain it.
- ii. To solve this problem, I had to start with small concepts and build upon it. For instance, it was foundational to know that the modulus operator performs integer division. Meaning, the remainder of the operation would

yield a digit. For example,  $4 \% 10 = 4$ . This was a start. My next step required me to find a way to progress through the digits of a number from right to left. Suppose I wanted to obtain the digit 2 in 24. If I were to perform  $24 \% 10$ , I would only get 4, which is not my goal. Therefore, one solution was to divide 24 by a power of 10 to move the decimal place. In this case, I would divide by  $10^1$  and then do  $\% 10$  to obtain 2. With this pattern, I was able to iterate through all 9 digits of the social security number.

b. Creating the buckets for Bucket sort

- i. To begin, I had difficulty deciding on what data structure to use to create the buckets. My initial thought was to use a 2D Array of integers. However, this was incorrect because Arrays are not dynamic. Therefore, distributing the elements into the buckets would not work since the length of the Arrays are fixed. My next attempt was to initialize an `ArrayList<Integer>` to size 10 to create the buckets. For each spot in the `ArrayList`, I also initialized a new `LinkedList`. Although I was able to distribute the elements in to the `LinkedLists`, my Insertion sort code was not compatible with `LinkedLists`. Initially, I tried to revise my Insertion sort code to accommodate `LinkedLists` but that required me to write a method to find a certain element and insert it. I felt like I was overcomplicating the procedure. Afterwards, I decided to create a main bucket with an `ArrayList<Integer>[]` and initialize each slot with an

`ArrayList<Integer>`. This was much simpler since I was able to write a compatible version of Insertion sort for `ArrayList<Integer>`.

- ii. Overall, the solution to this challenge was trial and error. At first, I did not have a strong sense of direction to pick the best data structure for this algorithm. But after trying out various methods and analyzing the outcomes, I was able to narrow down my results to an efficient choice.

## **F. Lessons Learned**

- a. Creating projects require careful planning. Although it is tempting to start coding right away, using extra time to think about each line of code is more productive. I noticed that when I started to code without structure, I got confused easily and had difficulty tracing my line of thought. Therefore, it is crucial to start planning with what the expectations of the program are. Not only does it form a concrete idea of what the results are supposed to be, but it provides a much more productive starting point. In addition, it is important to write down the conceptual steps. There have been several times when I expected a result and thought I had the steps correct, but realized that I had a few misconceptions.
- b. Starting early is always better than starting late. Although it is true that starting early may also mean finishing early, there is another crucial component. When I started early, I encountered my mistakes early on. Therefore, I had time to seek help and refine my thought process because learning does not always happen instantly. As a result, it may take a few days or a couple of tries at a problem for the concepts to register with the person.