Task : Radix sort

A screenshot of a game

Description automatically generated

Here is a simple explanation of how Radix Sort works with an example in C++:

1. **Find the maximum number** to know the number of digits.
2. **Create count arrays** for each digit. (For base 10 numbers, this means 10 arrays).
3. **Sort each digit**:
   * Do a count sort for each digit from least significant to most significant.
   * After all digits are sorted, the array is sorted.
4. **Identify the Range**: Determine the maximum value in the array to establish the range of possible values.
5. **Initialize Count Array**: Create an array (count array) with a length equal to the range, initialized to zero.
6. **Count Occurrences**: Go through the input array and increment the corresponding index in the count array.
7. **Compute Prefix Sum**: Transform the count array by adding the sum of the previous counts. This step determines the position of each element in the sorted array.
8. **Sort the Elements**: Iterate over the input array and place each element in its correct position in the output array, decrementing the count in the count array each time an element is placed.
9. **Copy Back**: Copy the sorted elements from the output array back to the original array.

#include <iostream>

using namespace std;

1. // get the max number from an array
2. int getMax(int arr[], int n) {
3. int max = arr[0];
4. for (int i = 1; i < n; i++) {
5. if (arr[i] > max) {
6. max = arr[i];
7. }
8. }
9. return max;
10. }
11. // Using counting sort to sort the elements
12. void countingSort(int arr[], int n, int place) {
13. int output[n];
14. int count[10] = {0};
15. for (int i = 0; i < n; i++) {
16. int index = arr[i] / place;
17. count[index % 10]++;
18. }
19. for (int i = 1; i < 10; i++) {
20. count[i] += count[i - 1];
21. }
22. for (int i = n - 1; i >= 0; i--) {
23. int index = arr[i] / place;
24. output[count[index % 10] - 1] = arr[i];
25. count[index % 10]--;
26. }
27. for (int i = 0; i < n; i++) {
28. arr[i] = output[i];
29. }
30. }
31. // Radix Sort
32. void radixSort(int arr[], int n) {
33. int max\_element = getMax(arr, n);
34. for (int place = 1; max\_element / place > 0; place \*= 10) {
35. countingSort(arr, n, place);
36. }
37. }
38. int main() {
39. int n;
40. cout << "Enter the number of elements: ";
41. cin >> n;
42. int arr[n];
43. cout << "Enter the elements: ";
44. for (int i = 0; i < n; i++) {
45. cin >> arr[i];
46. }
47. radixSort(arr, n);
48. cout << "Sorted array: "; //Assending order
49. for (int i = 0; i < n; i++) {
50. cout << arr[i] << " ";
51. }
52. return 0;
53. }