1. What is time complexity of LSD (Least Significant Digit) radix sort?
2. What is space complexity of LSD radix sort?
3. Is Radix Sort a stable sort? Why?
4. Give an application example for the linear-time sorting algorithm.
5. (1) Define what does “Sorting on several keys” mean.

(2) For the sorting algorithm on multiple keys, please propose an application example other than the sorting deck of cards in the textbook.

1. (1)We provide the LSD Radix Sort code below. Please explain what does the code of RadixSort do. (This code is provided in page 420 in the textbook “Fundamentals of Data Structures in C++”, 2nd)

template<class T>

int RadixSort(T \*a, int \*link, const int d, const int r, const int n){

    // Sort a[1:n] using a d-digit raidx-r sort. digit(a[i],j,r) returns the jth radix-r

    // digit(from the left) of a[i]'s key. Each digit is in the range is [0,r).

    // Sorting within a digit is done using a bin sort

    int e[r],f[r];

    int first=1;

    for(int i=0;i<n;i++) link[i]=i+1;

    link[n]=0;

    for(i=d-1;i>=0;i--){

        // Sort on digit i

        fill(f,f+r,0); // initialize bins to empty queues

        for(int current=first;current;current=link[current]){

            //put records into queue/bin

            int k=digit(a[current],i,r);

            if(f[k]==0) f[k]=current;

            else link[e[k]]=current;

            e[k]=current;

        }

        for(j=0;!f[j];j++); // Find first nonempty queue/bin

        first=f[j];

        int last=e[j];

        for(int k=j+1;k<r;k++){

            if(f[k]){

                link[last]=f[k];

                last=e[k];

            }

        }

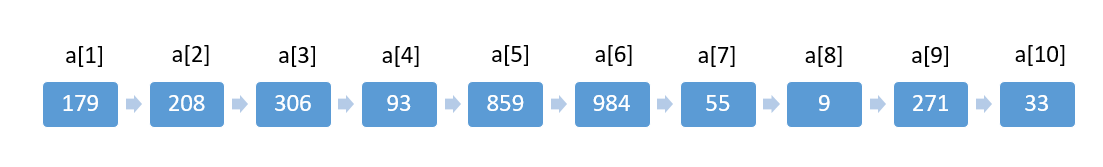
        link[last]=0;

    }

    return first;

}

You may use the following initial values of the a[] array elements to explain the algorithm.



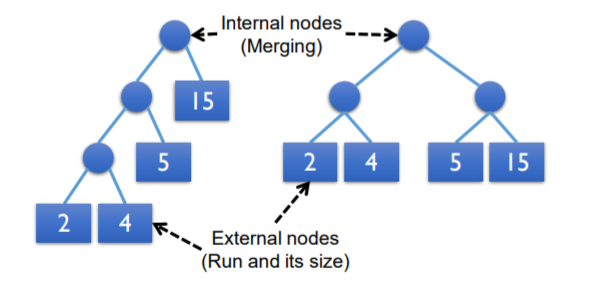
(2) Write the status of the list (12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18) at the end of each pass of RadixSort. Use r=10.

1. Please propose a hybrid sorting algorithm and discuss how and why the algorithm works, particular explain when and why one algorithm is switched to another one in the hybrid approach. You may go online and search for Hybrid Quicksort Algorithm (Quicksort+Insertion sort) to learn how people discuss the working principle of the algorithm.

Reference : <https://www.techiedelight.com/hybrid-quicksort/>

1. (1) What does “Optimize merge” mean?

(2) Why should we distinguish nodes into external nodes and internal nodes?



1. Please complete the following table and give examples to the worst and best time complexity cases. Explain why some have the same If best, average and worst time complexities.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Time Complexity | | |
|  | Best | Average | Worst |
| Insertion Sort |  |  |  |
| Quick Sort |  |  |  |
| Heap Sort |  |  |  |

1. Please give an example of the external sorting.
2. Introduce the concept and practical application of the algorithm.
3. Give a brief introduction to the implementation of the algorithm
4. Given an unsorted array, in which every element is at most *k* steps distance from its new position in the sorted array, where *k* is a positive integer smaller than the array size. Which sorting algorithm can be easily modified to fit the sorting criterion specified above and what is the obtainable time complexity?