Analysis of Cut Short Algorithm

Abstract--There are many sorting techniques which were developed over the hundreds of years and also optimized to reduce the complexity for worst and average cases. The name 'CutShort' signifies cutting of original array into shorter pieces. In this technique, the input array is divided into several subarray of shorter lengths based on bit-count, somewhat similar to the Bucket sort concept. Each sub-arrays contains those elements which can be represented by same number of bits..This technique reduces the complexity of worst and average case by a significant factor depending on the number of sub-arrays formed. This method can be used as a preprocessing technique and can be more beneficial if implemented in the processor as a subroutine.It can be used to optimize the runtime of various sorting algorithms.

Complexity Analysis

Best Case:

The best case occurs when the array is divided into several sub arrays of almost equal length. Thus, the length of input array is reduced to dmax times and processing time reduces by n log(dmax), here dmax is the maximum number of subarrays obtained from original array.

$n1 = n2 = n3 = n4 ....$

where $n1 + n2 + n3 + n4 + ... = n$

$T(n) = O(n1.\log n1 + n2.\log n2 + n3.\log n3 + ..)$

$T(n) = O((n1.\log n1) * dmax)$

$T(n) = O((n/ dmax) \log (n/dmax) * dmax)$

$T(n) = O(n \log (n/dmax))$

$T(n) = O(n \log (n) – n \log (dmax))$

Worst Case:

The worst case occurs when all the elements belongs to a single sub-array and the worst case complexity for optimised sorting method is O(n log n). So, in worst case, the time complexity will remain the same O(nlog n).

$T(n) = O(n.\log n)$

Average Case:

In the average case, consider that this technique have divided the array in atleast d different sections and on combining the factor d with the worst case complexity of traditional and optimal sorting techniques of O(n.log n), we find T(n) = O(n log n/d)

T(n) = O(n log (n) – n log(d))

where d is number of resulting sub-arrays.

Amortized analysis of Cut Short Algorithm

We used the Aggregate Analysis to calculate the Amortized cost of this algorithm. In this method we find an upper bound on the total sequence of n operations. The average cost per operation is called the Amortized cost.

Calculations:

- To read file, we require constant amount of time i.e. O(1).
- Bitmap array takes O (n logn) time as it has to convert n decimal values into binary numbers and get the bit count.
- The BitBand array takes constant time to rearrange the numbers with same BitCount in 1 group i.e. O(1).
- The quick sort algorithm takes O (n logn) in worst case.

Total time taken = O (1+nlogn+1+nlogn)

= O (nlogn)

Amortized cost Analysis:

Total no of operations:

- n decimal numbers to be converted into binary numbers would involve n arithmetic operations.
- To rearrange these n numbers would require n swaps in worst case.
- Quick sort involves the partition function which is the heart of the algorithm and it requires to n operations to be performed.

Total no of operation = n + n + n

= 3n

Amortized cost = Total time taken / Total no of operations

= O (n logn/3n)

= O (logn)