Quadruplet Sum Algorithm Analysis

Jake Imyak

February 26th, 2020

Run-time Analysis

Algorithm 1 Quadruplet Sum

```
1: procedure QuadSum(A[], n, sum)
2:     HashTable.init()
3:     for i ← 1 to n − 1 do
4:         for j ← i + 1 to n do
5:             value ← sum - (A[i] + A[j])
6:             if HashTable.Member(value) then
7:                 for Pair p ∈ HashTable.Member(value) do
8:                     x = Pair.x
9:                     y = Pair.y
10:                    if x is not i or j and y is not i or j then
11:                        Print x, y, i, j
12:                        return true
13:                    end if
14:                end for
15:            end if
16:            HashTable.Insert(A[i], A[j])
17:        end for
18:    end for
19:    return false
20: end procedure
```

The solution based goes through the list of numbers twice making the algorithm T(n) $\in \Theta(n^2)$.

The sum is subtracted from pair of numbers from indexes i and j from the array A. This value is used to see if we already have a pair of the same value to get the target sum in the hash table. We then retrieve that value from the hash table and then make sure their are no duplicates between the numbers. If no duplicates occur we print the numbers i, j, x, and y and return true. If not, we insert the pair into the hash table and continue to iterate through. Once the list has been traversed and no quadruplet sum has been found we return false.