Frederick Wittman
Lars Kotthoff
Rajiv Khadka
COSC 3015
09/27/19

1. **Code**

```
function divideAndConquerSum (a) {
 if (a.constructor == Array) {

   if (a.length == 0) {
     return 0;
   }

   if (a.length == 1) {
     return a[0];
   }

   else {
     return divideAndConquerSum(a.splice(0,1)) +
divideAndConquerSum(a.splice(0,1)) + divideAndConquerSum(a.splice(0));
   }
 }

 else {
 console.log("Function argument must be an array.");
 }
}
```

2. **T(n)**

   Let n be the number of elements in the array. In the base case, where n = 0, it is clear that T(0) = $\Theta(1)$. In general, T(n) = $\Theta(2)$ + T(n - 2). I solve the recurrence relation below.

   T(n) = $\Theta(2)$ + T(n - 2)

= 2 + T(n - 2)
= 2 + [2 + T(n - 4)]
= 4 + T(n - 4)
= 4 + [2 + T(n - 6)]
= 6 + T(n - 6)
= ...

It is clear that the number of operations increases linearly with n. k is a constant factor. Since we have T(0) = Θ(1) = 1, we allow n - k = 0.

= k + T(n - k)
= n + T(0)
= n + 1
= Θ(n)