Algorithm 1 Brute Force Method

Input: a - array of integers

Input: n - size of array

Output: sum - is the maximum sum of a contiguous subarray in a

```
procedure GS($a$)
    MaxSum ← a[0]                                   ▷ current maximum sum
    for i = 0 to n - 1 do
        for j = i to n - 1 do
            CurrentSum ← 0                          ▷ sum of subarray
            for k = i to j do
                CurrentSum ← CurrentSum + a[k]
            end for
            if MaxSum < CurrentSum then
                MaxSum ← CurrentSum                 ▷ update maximum sum
            end if
        end for
    end for
end procedure
```

Algorithm 2 Sliding Window Algorithm

Input: a - array of integers

Input: n - size of array

Output: sum - is the maximum sum of a contiguous subarray in a

```
procedure GS($a$)
    MaxSum ← a[0]                                   ▷ current maximum sum
    for i = 0 to n - 1 do
        CurrentSum ← 0
        for j = i to n - 1 do
            CurrentSum ← CurrentSum + $a[j]$;
            if MaxSum < CurrentSum then             ▷ update maximumx sum
                MaxSum ← CurrentSum
            end if
        end for
    end for
end procedure
```

Algorithm 3 Kadane's Algorithm

```
Input: $a$ - array of integers
Input: $n$ - size of array
Output: $sum$ - is the maximum sum of a contiguous subarray in $a$
procedure GS($a$)
    MaxSum ← a[0]                                    ▷ current maximum sum
    CurrentSum ← MaxSum
    for i = 0 to n - 1 do
        CurrentSum ← CurrentSum + $a[i]$
        if CurrentSum < a[i] then        ▷ current sum up to this less than current point
            CurrentSum ← $a[i]$
        end if
        if MaxSum < CurrentSum then                  ▷ update maximumx sum
            MaxSum ← CurrentSum
        end if
    end for
end procedure
```

Algorithm 4 Brute Force Method

```
Input: $x$ - decision tree
Output: $abc$ is $x$ in Txx Bxxxxx Gxxxx
procedure Recursion($a$)
    $a \leftarrow$ Recursion($a$)                    ▷ Call Recursion again
    return $a$
end procedure
```

Algorithm 5 Euclid's algorithm

```
procedure Euclid($a, b$)                             ▷ The g.c.d. of a and b
    $r \leftarrow a \bmod b$
    while $r \neq 0$ do                              ▷ We have the answer if r is 0
        $a \leftarrow \sum_{i=1}^{n} x_i$            ▷ Nonsense to show that tall lines might work
        $a \leftarrow b$
        $b \leftarrow r$
        $r \leftarrow a \bmod b$
    end while
    return $b$                                       ▷ The gcd is b
end procedure
```