1 Pseudocode Algorithm

Algorithm 1 Find max two such that:

$$i < j$$

and

$$A[i] \leq A[j]$$

```
function FindMax(A, p, r)
    if p equals r then return (−∞, −∞)
    else
        mid ← ⌊(p + r)/2⌋
        leftSolution ← FindMax(A, p, mid)
        rightSolution ← FindMax(A, mid + 1, r)
        crossingSolution ← FindCrossingMax(A, p, mid, r)
        return Max(leftSolution, crossingSolution, rightSolution)
    end if
end function
```

Algorithm 2 Perform work to actually find the max two for a given p and r.

```
function FindCrossingMax(A, p, mid, r)
    i ← mid
    j ← mid + 1
    max ← A[i] + A[j]
    for k from mid down to p do
        if max < A[k] + A[j] then
            i ← k
            max ← A[k] + A[j]
        end if
    end for
    for k from mid + 1 up to r do
        if max < A[k] + A[i] then
            j ← k
            max ← A[k] + A[i]
        end if
    end for
    return (i, j)
end function
```

2 Complexity Analysis

We can think of this algorithm in terms of three parts: left call, right call and a function that finds the answer which crosses over the from the left to the right.

With that in mind, the time complexity of this algorithm can be modeled using the following equation:

$$\mathrm{T}(n) = \mathrm{T}(\lfloor n/2 \rfloor) + \mathrm{T}(\lceil n/2 \rceil) + \mathrm{F}(n) + \mathcal{O}(1)$$

Recognizing that the time complexity is monotonically increasing and is bounded by the functions and not the constant $\mathcal{O}(1)$, we can eliminate the floor and ceiling functions and simplify the constant time element as follows:

$$\begin{aligned}\mathrm{T}(n) &= \mathrm{T}(n/2) + \mathrm{T}(n/2) + n + 1 \\ &= 2 \cdot \mathrm{T}(n/2) + n + 1\end{aligned}$$

Using the master method it is evident that

$$\begin{aligned}a &= 2 \\ b &= 2 \\ \mathrm{F}(n) &= n\end{aligned}$$

Therefore

$$\begin{aligned}\mathrm{F}(n) &= \theta(n^{log_b a}) \\ &= \theta(n^{log_2 2}) \\ &= \theta(n^1) \\ &= \theta(n)\end{aligned}$$

Implying that we should use rule 2 of the master method. Using rule two, we determine the following:

$$\begin{aligned}\mathrm{T}(n) &= \theta(n^{log_b a} \cdot log_2 n) \\ &= \theta(n \cdot log_2 n)\end{aligned}$$