CS2700 : Programming and Data Structures.

WEEK 3 : Arrays

Goal for Today :

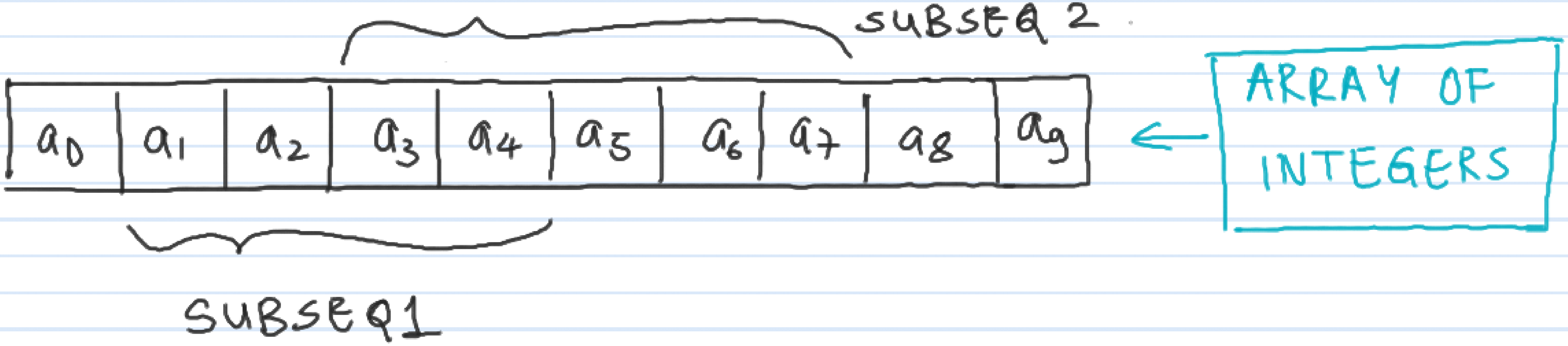
Maximum Subsequence Problem

(1) Design algorithm

(2) Analyze

(3) Is this the best possible?

MAX SUBSEQUENCE PROBLEM.



VALUE OF A SUBSEQ $a_i \cdots a_j$: $\sum_{k=i}^{j} a_k$

GOAL : FIND A SUBSEQ OF MAX VALUE.

MAX SUBSEQ EXAMPLE.

0	1	2	3	4	5	6
1	2	3	-4	5	-8	7

What is the max subseq in this example?

Some assumptions

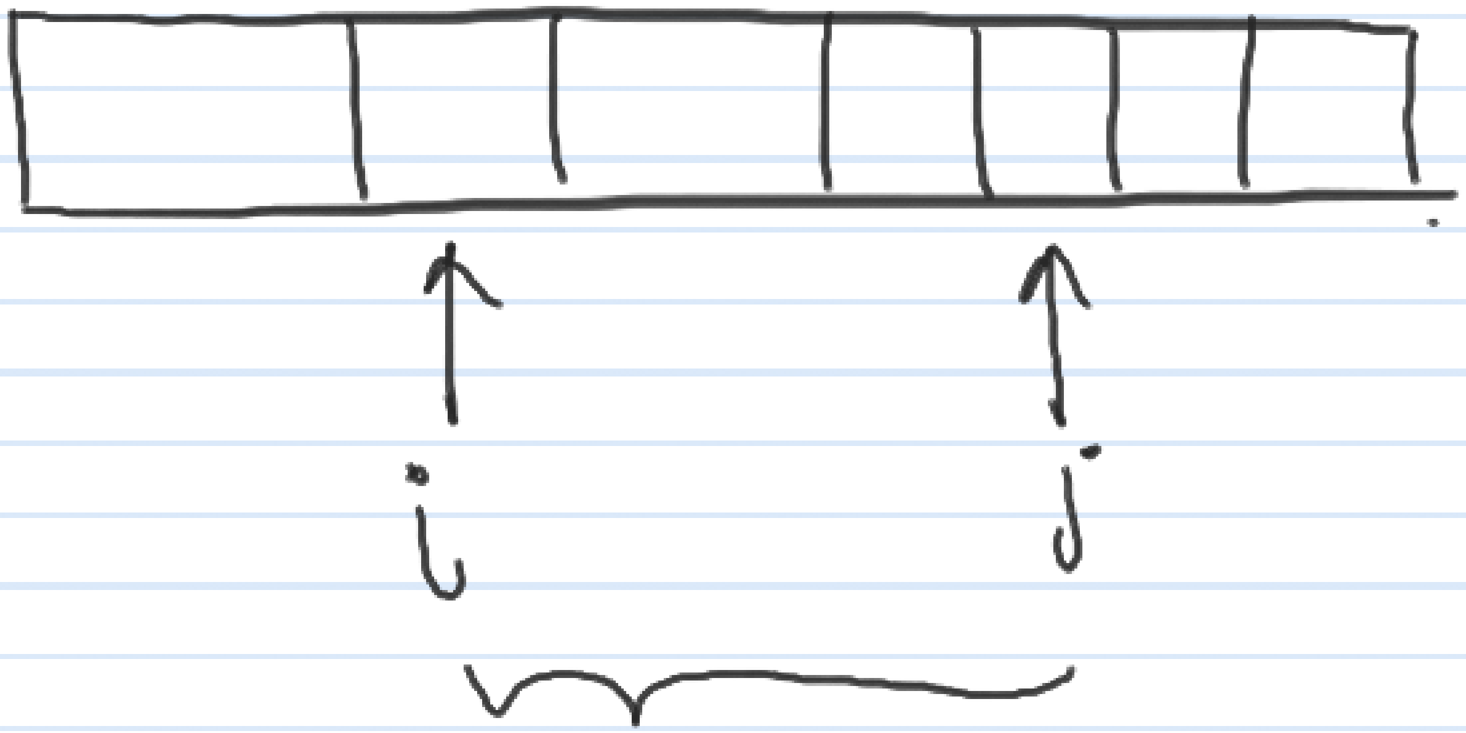
(+) ve and (−) ve values are present

answer is non-negative

MAX SUBSEQ ALGO 1 (FIRST ATTEMPT)

1) How many subsequences?

2) Once I have a subsequence, compute the value.



MAX SUBSEQ ALGO 1 (PSEUDOCODE)

```
maxsum = 0;
for i = 0 to n-1 {
    for j = i to n-1 {
        currsum = 0;
        for k = i to j {
            currsum = currsum + A[k];
        }
        if currsum > maxsum
            maxsum = currsum
    }
}
```

ALGO 1 : ANALYSIS.

```
for i = 0 to n-1
    for j = i to n-1
        for k = i to j
            { }
```

$O(n^3)$

$\Theta(n^3)$

$$\sum_{i=0}^{n-1} \sum_{j=i}^{n-1} \sum_{k=i}^{j} 1$$

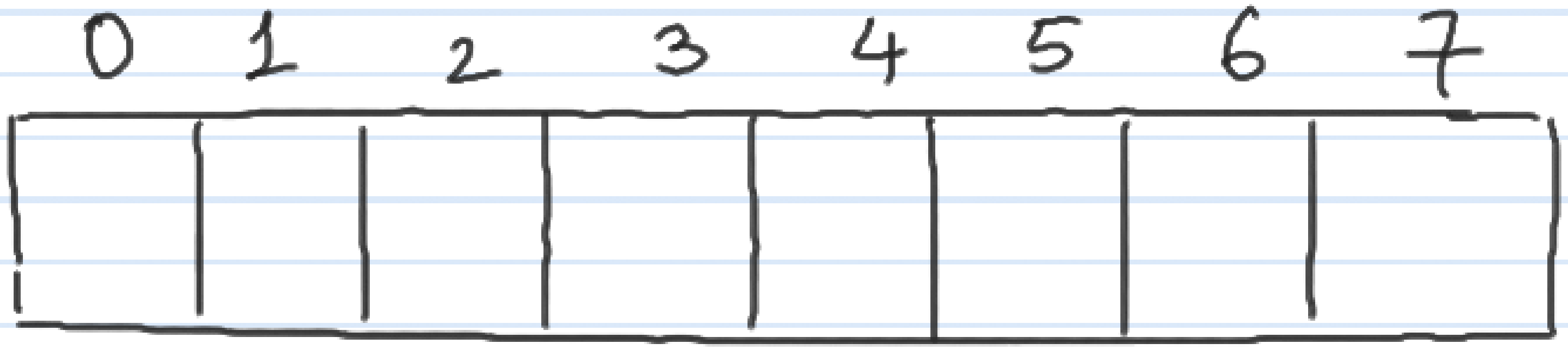
$$\sum_{i=0}^{n-1} \sum_{j=i}^{n-1} (j-i+1)$$

$1+2+3 \cdots (n-i)$

$$\sum_{i=0}^{n-1} \frac{(n-i+1)\ (n-i)}{2}$$

$\rightarrow (n^3+3n^2+2n)/6$

MAX SUBSEQ: ALGO2



$i = 2$ $j = 5$

$i = 2$ $j = 6$

$i = 2$ $j = 7$

for the same value of i but different values of j. reuse computation

MAX SUBSEQ : ALGO2

```
maxsum = 0;
for i = 0 to n-1 {
    currsum = 0;
    for j = i to n-1 {

        currsum = currsum + A[j]

        if currsum > maxsum

            maxsum = currsum;

    }
}
```

ANALYSIS.

SHOW THAT ALGO is $\Theta(n^2)$

MAX SUBSEQ ALGO 3

0	1	2	3	4	5	6	7
7	-2	8	10	-20	-12	13	16

B[i] = value of max subsequence ending at i

(if value of subseq is positive else store B[i] = 0

	0	1	2	3	4	5	6	7
B	7	5	13	23	3	0	13	29

$B[i] = \max(0, B[i-1] + A[i])$

$B[0] = \max(0, A[0])$

where is the final answer?

max value in B

MAX SUBSEQ ALGO3

```
B[0] = max(0, A[0])
for i = 1 to n-1 {

    B[i] = B[i-1] + A[i]
    if B[i] < 0
        B[i] = 0;
}
find max(B)
```

EXAMPLE

A

0	1	2	3	4	5	6	7
7	-2	8	10	-20	-12	13	16

TODOS:

(1) ARGUE CORRECTNESS.

(2) IS AUX STORAGE array B required?

SUMMARY OF MAX SUBSEQ PROBLEM.

- $\Theta(n^3)$, $\Theta(n^2)$ and finally $\Theta(n)$ time algos.
- We did not study $O(n \log n)$ algo using divide and conquer.
- Efficient algo was simpler to code ($\Theta(n)$ requires a clever idea
- Efficient algo performs significantly faster

PROBLEM 2 : MISSING VALUE PROBLEM

INPUT: ARRAY OF n INTEGERS (unsorted)

GOAL: FIND SMALLEST POSITIVE INTEGER MISSING.

A

-2	1	14	-7	3	4	-6	15	-8

missing value = 2

A'

1	2	3	4	5	6

missing value = 7

PROBLEM2 : MISSING VALUE PROBLEM.

1) Search for $i = 1, 2, 3 \ldots$ linearly. How many values to search.?

2) Sort array then use binary search.

3) Maintain an aux array of size $1 \ldots max$

not linear time

Use linear scan of array A.

4) Maintain an aux array of smaller size.

$(n+1)$ suffices → why?

PROBEM2 : MISSING VALUE PROBLEM.

A : input array of size n containing positive & negative integers (unsorted)

- B : aux array indexed from 1 .. n+1
- $B[j] = 0$ for all $j = 1, \ldots, n+1$
- Scan A linearly {

  $k = A[i]$;

  if $k > 0$ and $k \leq n+1$ $B[k] = 1$

}

The first index j in B s.t $B[j] == 0$ is the smallest pos missing elem in A.