EECS2011 Fundamentals of Data Structures (Winter 2022)

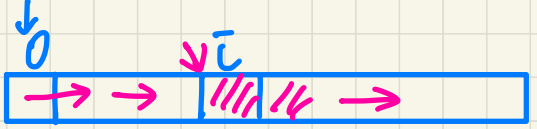
Q&A - Week 2 Lecture

Wednesday, January 26

Announcements

- Lecture W3 released (SLL)
- Assignment 1 (requiring SLL) to be released on Monday.
- Plan of Returning In-Person (starting Feb. 14)
  + Unchanged
    * Pre-recorded lectures
    * Zoom Weekly Q&A and Office hours **in the first instance**
    * Online Programming & Written tests **in the first instance**
  + Changed
    * In-Person Exam
  + To be determined:
    * Some (programming and/or written) tests **may be** in-person, in which case you'll be notified **at least one week** in advance.

Inserting into an Array




```
String[] insertAt(String[] a, int n, String e, int i)
  String[] result = new String[n + 1];
  for(int j = 0; j <= i - 1; j ++){ result[j] = a[j]; }
  result[i] = e;
  for(int j = i + 1; j <= n; j ++){ result[j] = a[j-1]; }
  return result;
```

For the second loop, what's the worst case of the number of iterations?

- [i + 1, n]
- insertAt(a, a.length, e, 0)

$n - (i+1) + 1 = n - i$ → # iterations

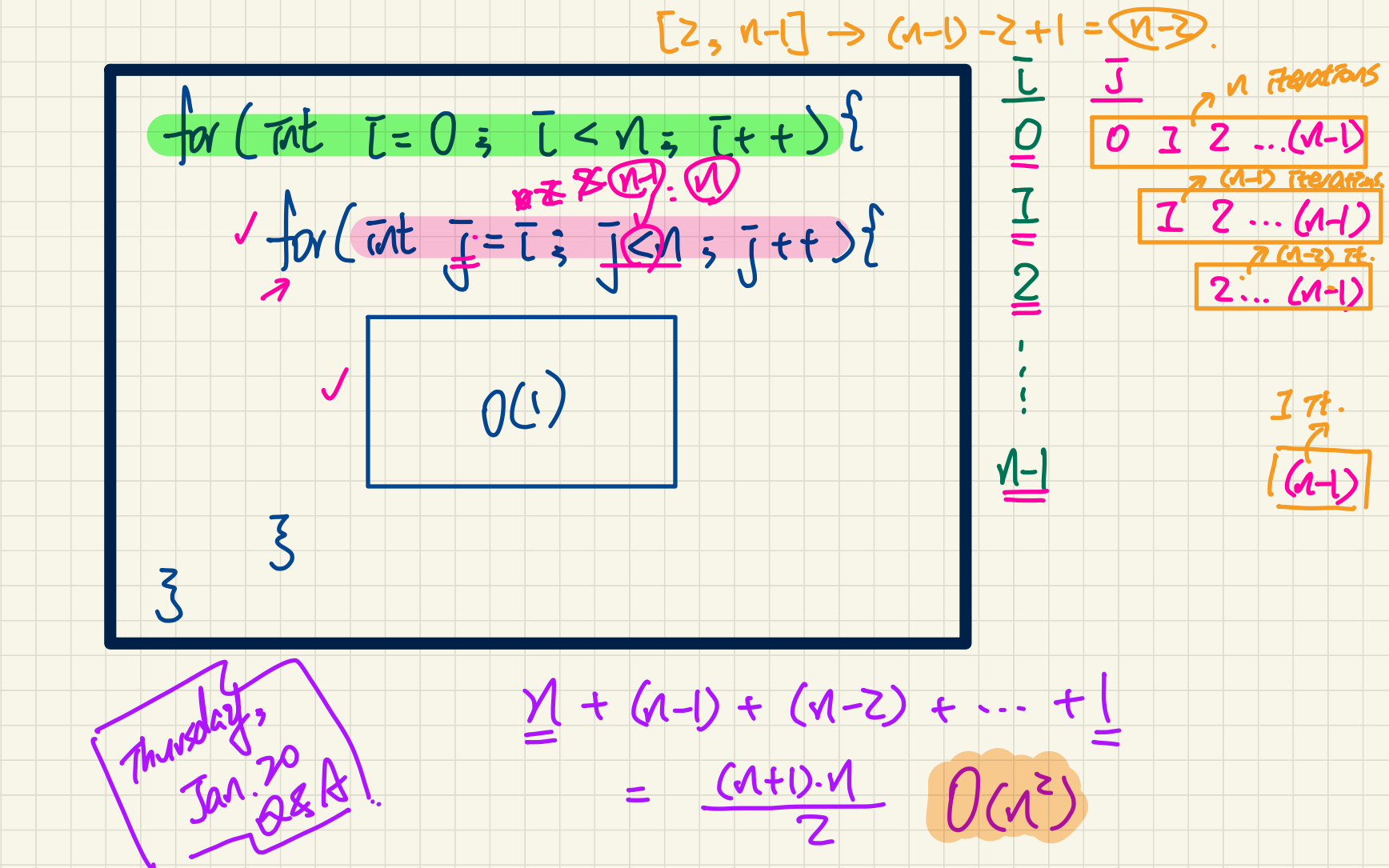
to maximize this #

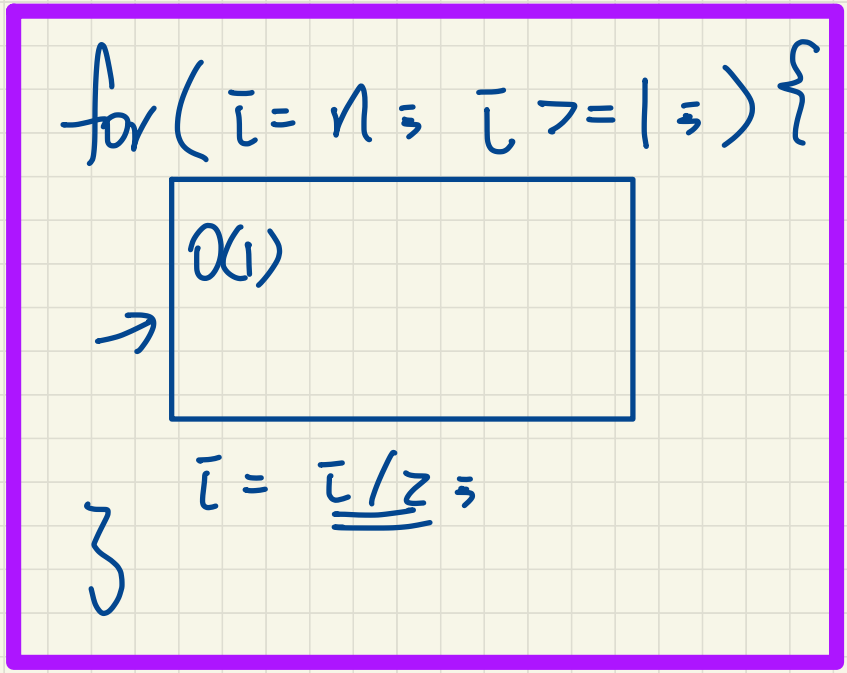
↓

i is minized

↓

i=0


[2, n-1] → (n-1) - 2 + 1 = n-2.
for (int i = 0; i < n; i++) {
for (int j = i; j < n; j++) {
O(1)
}
}
i
0
1
2
...
n-1
j
n iterations
0 1 2 ... (n-1)
(n-1) iterations
1 2 ... (n-1)
(n-2) it.
2 ... (n-1)
1 it.
(n-1)
n + (n-1) + (n-2) + ... + 1
= (n+1)·n / 2
O(n²)
Thursday, Jan. 20 Q&A



n = 1024

i	iteration
1024	1
512	2
256	3
128	4
64	5
32	6
16	7
⋮	⋮
1	10

$\log_2 1024$ = "10"

Alternative Solution to groupSum

Credit: Mohammad P.

```
public boolean groupSum(int start, int[] nums, int target) {
  if (start >= nums.length) {
    return target == 0;
  }
  else {
    return groupSum(start+1, nums, target-nums[start]) ||
      groupSum(start+1, nums, target);
  }
}
```

Problem on Recursion: Climbing Staircase

climb
steps
staircase

You are asked to program this method:

public **int** climb(**int** h, **int** n)

The return value suggests the number of possible ways
for climbing a stair of height h, while each climb is up to n steps.

Assumptions: n ≤ h, each climb takes at least 1 step

e.g., climb(4, 2) returns 5, meaning that there are 5 ways
for climbing a stair of height 4 and each climb takes up to 2 steps.

1. 1 step, 1 step, 1 step, 1 step
2. 1 step, 1 step, 2 steps
3. 1 step, 2 steps, 1 step
4. 2 steps, 1 step, 1 step
5. 2 steps, 2 steps

first climb: 1 step

first climb: 2 steps

climb(4-1, 2) = 3

climb(4-2, 2) = 2

1, 2, ..., n

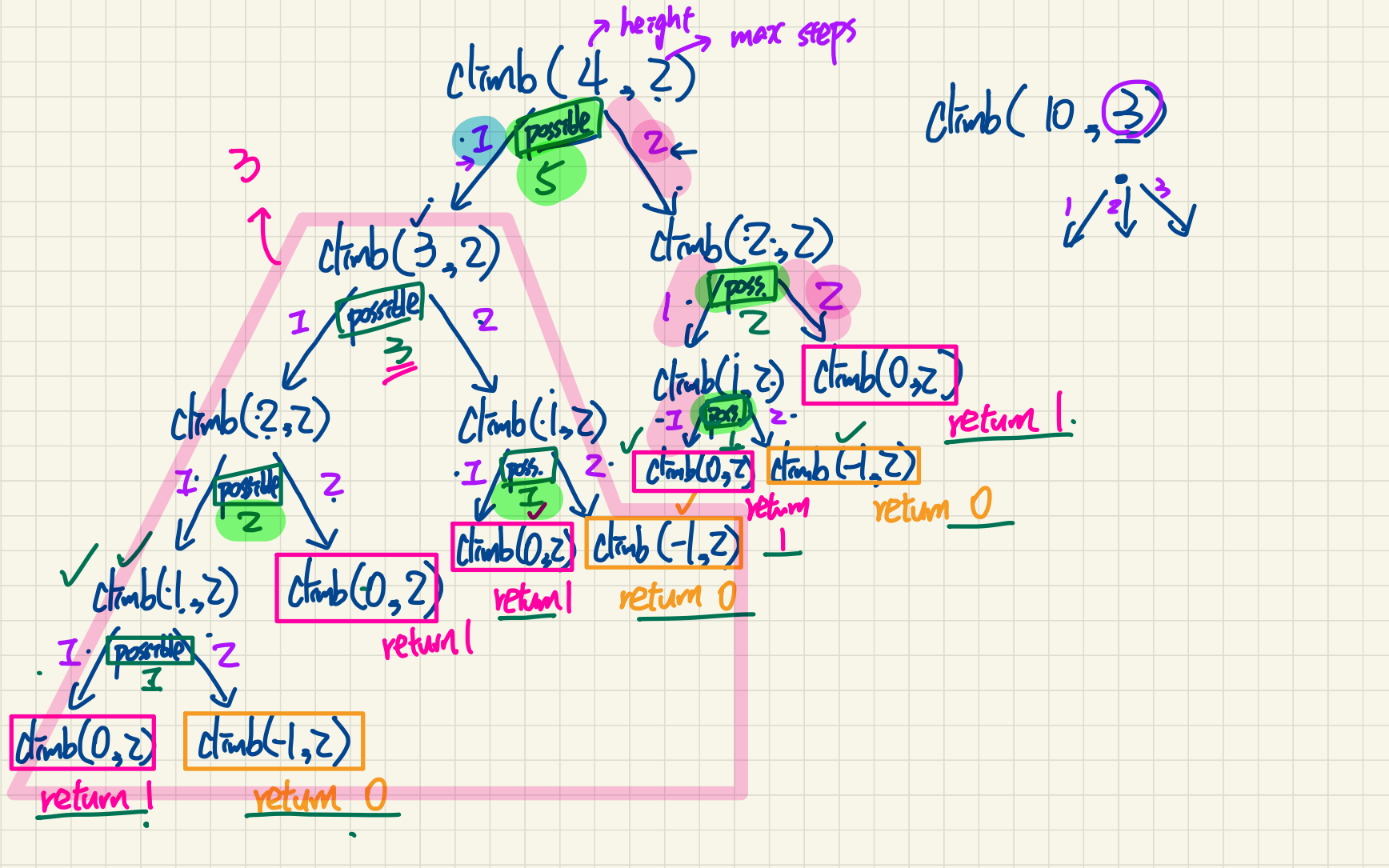
climb(h, n)

1 step, climb(h-1, n)

2 steps, climb(h-2, n)

i steps, climb(h-n, n)

strictly smaller prob.


height
max steps
climb(4,2)
1
2
possible
5
climb(3,2)
climb(2,2)
1
possible
2
2
climb(1,2)
possible
1
2
climb(1,2)
possible
2
2
climb(0,2)
return 1
climb(-1,2)
return 0
climb(0,2)
return 1
climb(-1,2)
return 0
climb(0,2)
return 1
climb(-1,2)
return 0
climb(10,3)
1
2
3