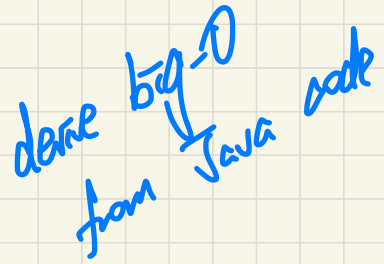
EECS2011 Fundamentals of Data Structures (Winter 2022)

Q&A - Week 1 Lecture

Thursday, January 20

Announcements

- Lecture W2 released
- Background Study on Java Generics
- Q&A materials from yesterday



Problem: Counting the Number of Primitive Operations

↳ unnecessary to understand the logic before starting to count

```
1  boolean containsDuplicate (int[] a, int n) {
2    for (int i = 0; i < n; ) {
3      for (int j = 0; j < n; ) {
4        if (i != j && a[i] == a[j]) {
5          return true; }
6        j ++; }
7      i ++; }
8    return false; }
```

→ 8

4n+4

J	J<n
0	[0, n-1] ↳ (n-1)-0+1
1	True → n ↳ iterations
⋮	
n-1	
n	→ F n<n → False → exit

Q1. How many times "j<n" executed? n+1
Q2. How many times BOL executed? n
Q3. How many times "<n" executed? n+1
Q4. How many times BOL executed? n

POs in each it. of outer loop: (4n+2) + 2 = 4n+4
(4n+2): inner loop; 2: i++

POs in exec. outer loop: $1 + \underset{i=0}{(n+1)} + \underset{i<n}{n \cdot (4n+4)} = 4n^2+5n+2$

POs in each iteration of inner loop: 8

POs in exec. inner loop: $1 + \underset{j=0}{1} \cdot \underset{j<n}{(n+1)} + n \cdot 8 = 4n+2$

#s in [a, b] = b-a+1

[1, 10] = 10-1+1 = 10

Overall: $(4n^2+5n+2) + \underset{\text{rtn. false}}{1} = 4n^2+5n+3$

char charAt(int i)

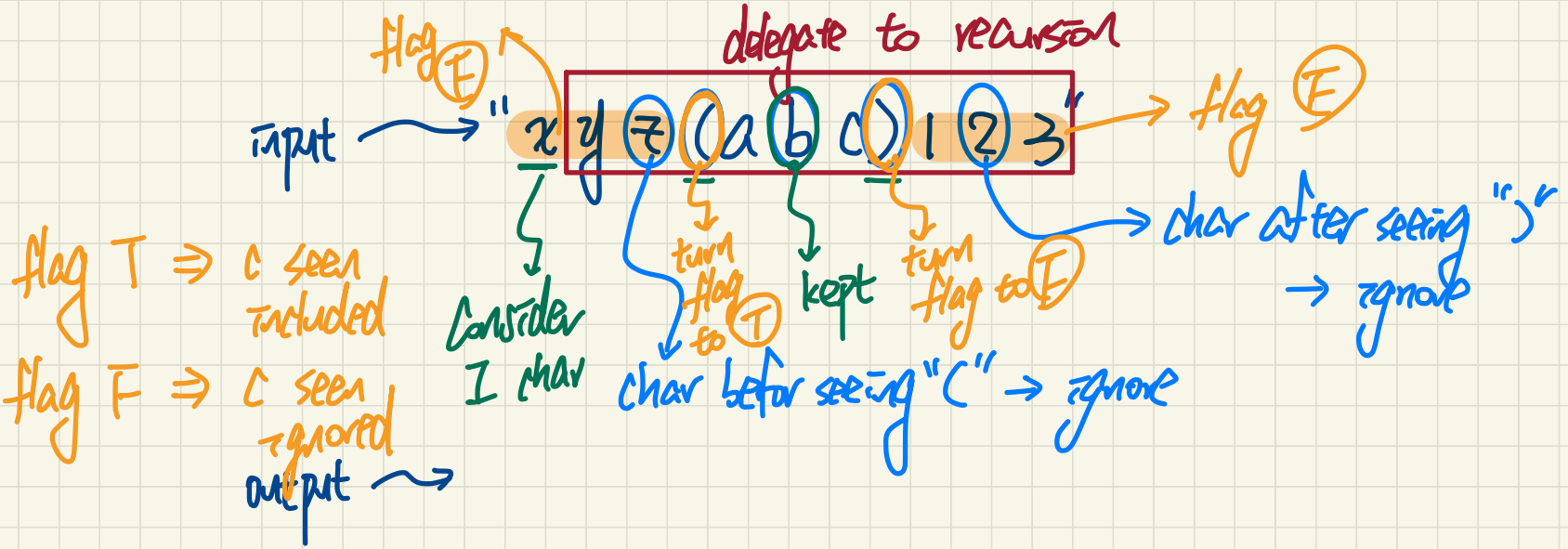
Problem on Recursion

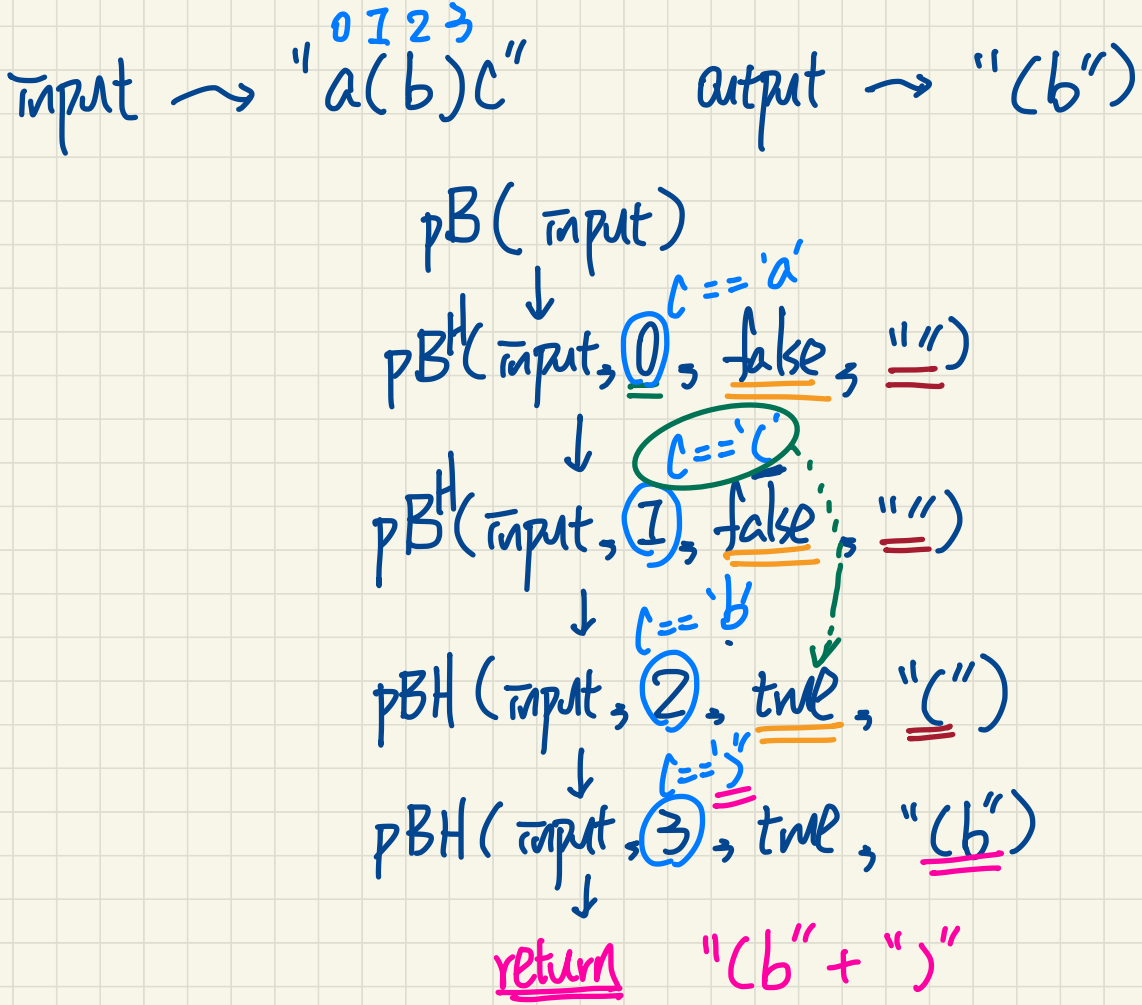
https://codingbat.com/prob/p137918

Given a string that contains a single pair of parenthesis, compute recursively a new string made of only of the parenthesis and their contents, so "xyz(abc)123" yields "(abc)".

parenBit("xyz(abc)123") → "(abc)"
parenBit("x(hello)") → "(hello)"
parenBit("(xy)1") → "(xy)"

Assumption: single pair of ( ).




input → "a(b)c"
0 1 2 3
output → "(b"
pB(input)
pBH(input, 0, false, "")
c == 'a'
pBH(input, 1, false, "")
c == '('
pBH(input, 2, true, "(")
c == 'b'
pBH(input, 3, true, "(b")
c == ')'
return "(b" + ")"