Data Structure

Lab Session 3:
Recursion Algorithms

U Kang

Seoul National University

Goals

- Learn how to write recursive functions
  - Find a base case (or terminate condition)
  - Find recurrence relations
- Implement two recursion algorithms
  - Greatest Common Divisor (recursive version)
  - Pascal's Triangle (recursive version)

Notice

- After implementing each algorithm, you have to check if your program works well.
  - Download sample input and output from the eTL.
  - Make a jar file and test your program by using it.
    - See the slides from the 1st lab session to check how to make a jar file
- Please raise your hand and ask to the T.A.s if you have a problem while implementing it.
- You need to stay for at least an hour.

Java Version

- **We will grade your future programming assignments with only Java 11.0.4**
  - Normally all versions starting with 11 are compatible with Java 11.0.4 but double check your jar file using a computer with Java 11.0.4 installed to be sure
    - Ex) Java 11.0.2, Java 11.0.3
- You can check your java version using the following command in the terminal(cmd)
  - `java --version`

```
➜ java --version
java 11.0.4 2019-07-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.4+10-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.4+10-LTS, mixed mode)
```

Import projects

- Download the skeleton projects for each algorithm from the eTL
- Extract the project, and import it into IntelliJ
  - See the slides of 1st lab session to check how to import the project in IntelliJ.

Greatest Common Divisor (GCD)

- GCD of two or more natural numbers is the largest positive integer that divides the integers without remainder.
  - For example, the GCD of 8 and 12 is 4.
- The Euclidean algorithm is a simple algorithm to calculate the GCD of two natural numbers:
  - $\gcd(a, 0) = a$
  - $\gcd(a, b) = \gcd(b, a \bmod b)$

I/O Specification (GCD)

Input form	Output form
`(a) (b)`	`GCD<(a), (b)> = (GCD of a and b)`

Description

- (a) and (b) are the natural numbers and divided by a space.
- Print the GCD of given numbers for each input.

Example Input	Example Output
`8 20`	`GCD<8, 20> = 4`

Sample Input and Output (GCD)

Sample input

```
7 9
4 24
3 9
48 132
169 156
```

Sample output

```
GCD<7, 9> = 1
GCD<4, 24> = 4
GCD<3, 9> = 3
GCD<48, 132> = 12
GCD<169, 156> = 13
```

Pascal's Triangle (1)

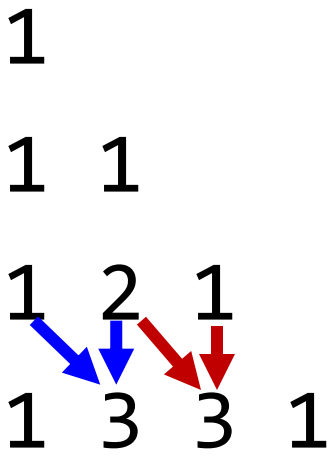
- Pascal's triangle is a triangular array of the binomial coefficients.

	0	**1**	**2**	**3**	**4**
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

- The entry in n-th row and k-th column of the triangle is denoted by $\binom{n}{k}$.

Pascal's Triangle (2)

- There is a recurrence relation between entries in the triangle:
  - $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- Each entry in the triangle is the sum of the two entries above.



I/O Specification (Pascal's Triangle)

- binomial

Input form	Output form
`binomial (n) (k)`	`nCk = (the entry in (n, k))`

Description

- Given $n \geq k \geq 0$, print $\binom{n}{k}$.
- You don't need to check range of n and k.

Example Input	Example Output
`binomial 5 3`	`5C3 = 10`

I/O Specification (Pascal's Triangle)

- draw

Input form	Output form
`draw (n)`	`Pascal's Triangle ((n))` `(draw the triangle)`
Description	
- Given $n \geq 0$, print the Pascal's triangle with $n + 1$ rows	
Example Input	**Example Output**

`draw 2`	`Pascal's Triangle (2)`
	`1`
	`1 1`
	`1 2 1`

Sample Input and Output (Pascal's Triangle)

Sample input

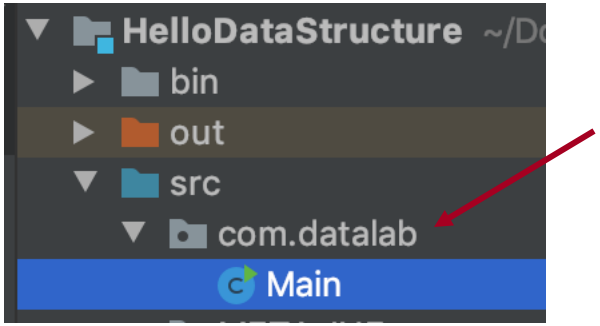
```
binomial 7 3
binomial 4 2
draw 4
```

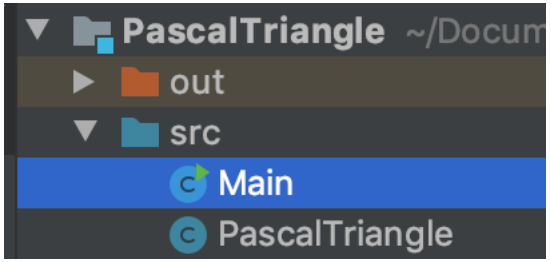
Sample output

```
7C3 = 35
4C2 = 6
Pascal Triangle (4)
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Execute Jar File

- Lab 01
  - `java -classpath <jarFileName.jar> com.datalab.Main`
- This session
  - `java -classpath <jarFileName.jar> Main`
  - `/com` and `/datalab` directories does not exist in the `/src`





Questions?

Course Information

- T.A.
  - Office 301-B119
  - Huiwen Xu
    - E-mail: xuhuiwen33@snu.ac.kr
    - Office hour: Wed 13:00 – 14:00
  - Junghoon Kim
    - E-mail: joseph.junghoon.kim@gmail.com
    - Office hour: Thu 16:00 – 17:00
  - Seungcheol Park
    - E-mail: ant6si@snu.ac.kr
    - Office hour: Thu 14:00 – 15:00
  - Chaeheum Park (Lead T.A. for this session)
    - E-mail: chaeheum@snu.ac.kr
    - Office hour: Tue 14:00 – 15:00