## CS1110 30 November 2010 Ragged arrays, Applets

Reading for today: sec. 9.3. chapter 16, applets

Type of d is **int**[][]

("**int** array array"/ "an array of **int** arrays")

To declare variable d:

**int** d[][];

To create a new array and assign it to d:

d= **new int**[5][4]; why not int[5, 4]?

or, using an array initializer,

d= **new int**[][]{ {5,4,7,3}, {4,8,9,7}, {5,1,2,3}, {4,1,2,9}, {6,7,8,0} };

| d | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 5 | 4 | 7 | 3 |
| 1 | 4 | 8 | 9 | 7 |
| 2 | 5 | 1 | 2 | 3 |
| 3 | 4 | 1 | 2 | 9 |
| 4 | 6 | 7 | 8 | 0 |

Some mysteries: an odd asymmetry, and strange toString output (see demo).

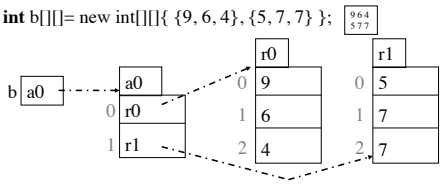
Number of rows of d: d.length

Number of columns in row r of d: d[r].length



## How multi-dimensional arrays are stored: arrays of arrays

**int** b[][]= new int[][]{ {9, 6, 4}, {5, 7, 7} };




b holds the name of a one-dimensional array object with b.length elements; its elements are the names of 1D arrays.

b[i] holds the name of a 1D array of **int**s of length b[i].length

java.util.Arrays.deepToString recursively creates an appropriate String.



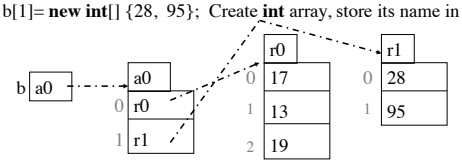
## Ragged arrays: rows have different lengths

**int**[][] b; Declare variable b of type **int**[][]

b= **new int**[2][] Create a 1-D array of length 2 and store its name in b. Its elements have type **int**[] (and start as **null**).

b[0]= **new int**[] {17, 13, 19}; Create **int** array, store its name in b[0].

b[1]= **new int**[] {28, 95}; Create **int** array, store its name in b[1].






## Pascal's Triangle

Row r: r+1 values. First, last entries on row are 1. Others are sum of the two above

| | |
|---|---|
| 1 | 0 |
| 1 1 | 1 |
| 1 2 1 | 2 |
| 1 3 3 1 | 3 |
| 1 4 6 4 1 | 4 |
| 1 5 10 10 5 1 | 5 |
| … | |

p[i][j] = "i choose j" = $\binom{i}{j}$ number of ways j elements can be chosen from a set of size i

recursive formula: for 0 < i < j, p[i][j] = p[i–1][j–1] + p[i–1][j]



## Pascal's Triangle

| | |
|---|---|
| 1 | 0 |
| 1 1 | 1 |
| 1 2 1 | 2 |
| 1 3 3 1 | 3 |
| 1 4 6 4 1 | 4 |
| 1 5 10 10 5 1 | 5 |

Binomial theorem: Row r gives the coefficients of $(x + y)^r$

$(x + y)^2 = 1x^2 + 2xy + 1y^2$

$(x + y)^3 = 1x^3 + 3x^2y + 3xy^2 + 1y^3$

$(x + y)^r = \sum_{0 \le k \le r}$ (k choose r) $x^k y^{r-k}$



## Function to compute first r rows of Pascal's Triangle in a ragged array

```
/** = ragged array of first n rows of Pascal's triangle.
    Precondition: 0 ≤ n */
public static int[][] pascalTriangle(int n) {
    int[][] b= new int[n][];     // First n rows of Pascal's triangle
    // invariant: rows 0..i-1 have been created
    for (int i= 0; i != b.length; i= i+1) {
      // Create row i of Pascal's triangle
        b[i]= new int[i+1];

      // Calculate row i of Pascal's triangle
        b[i][0]= 1;
        // invariant b[i][0..j-1] have been created
        for (int j= 1; j < i; j= j+1) {
          b[i][j]= b[i-1][j-1] + b[i-1][j];
        }
        b[i][i]= 1;
    }
    return b;
}
```

Revealing comments about A6

1. Introduce methods to make programming easier and less time-consuming

Note the precise specification

```
/** Hide n in pixel no. p
    of currentIm.
    Pre: 0 <= n < 1000. */
private void hide(int n,
                  int p)
```

```
/**  Hide m in this image ...
     Return ...     */
public boolean hide(String m) {
  Check length …

  hide('\t', 0);
  hide(m.length()/1000, 1);
  hide(m.length()%1000, 2);
  hide('\t', 3);

  int p= 4;
  // inv: tab char, mess. length, tab
  //      char, and pixels 0..k-1 hidden ...
  for (int k= 0; k < m.length(); k= k+1) {
      hide(m.charAt(k), p);
      p= p+1;
  }
  return true;
}
```

Test it before writing reveal!



1. Slow to reveal!

```
/** Extract and return … */
public String reveal() {
  …

  int p= 4;
  String result= "";

  // inv: All hidden chars before
  // pixel p are in result[0..k-1]
  for (int k= 0; k < len; k= k+1) {
     result= result +
           (char) (getHidden(p));
     p= p+1;
  }
  return result;
}
```

gives $n^2$ algorithm (n is message length)

```
/** Extract and return … */
public String reveal() {
  …

  int p= 4;
  char[] result= new char[len];

  // inv: All hidden chars before
  // pixel p are in result[0..k-1]
  for (int k= 0; k < len; k= k+1) {
     result[k]=
           (char) (getHidden(p));
     p= p+1;
  }
  return new String(result);
}
```

linear algorithm



**Applet: a java program that can be called from a web page (in your browser)**

```
public class C {
   public static void main(String[] args)
      { … }
}
```
application

```
import javax.swing.*;
public class A extends JApplet {
   public void init() { … }
   public void start() { …}
   public void stop() { … }
   public void destroy() { … }
}
```
applet

**Four inherited procedures:**

- called to initialize
- called to start processing
- called to stop processing
- called to destroy resources (just before killing the applet)



```
public class Quizit extends JApplet {
  // = "started as an applet"
  private boolean isApplet= false;

  public Quizit() {}

  /** = "started as an applet" */
  public boolean isApplet()
    { return isApplet; }

  public static void main(
                    String[] pars) {
     Quizit a= new Quizit();
     a.isApplet= false;  …
     a.readTopicsFile(br);
     a.gui= new A7GUI(
                a.fillItems(), a);
  }
```

**Quizit is both an applet and an application**

```
/** initialize applet */
public void init() {
    isApplet= true;  …
    readTopicsFile(br);
    gui= new A7GUI(
         fillItems(), this);
  }
```



**An html (HyperText Markup Language) file**

```
<html>
  <head>  <title>Just a title</title>  </head>
  <body>
    <p align="center"><B>Demo Links and Images</i> </p>
    <p>This is
      <a href="http://www.cs.cornell.edu/courses/cs1110/2009sp/"> a link</a></p>
    <p>This <a href="http://www.cs.cornell.edu/courses/cs1110/2009sp/"
                    target="_blank">link</a>
       opens a new window</p>

    <p>Below  is an image </p>

    <p><img src="Hilbert_curve.png"
                    width="544"
                    height="352">
    </p>
  </body>
</html>
```

tags

| | |
|---|---|
| **<html>** | start an html page |
| **<head>** | start the "heading" |
| **<title>** | the title for the page |
| **<body>** | start the body, content, of the page |
| **<p>** | begin a paragraph |
| **<a>** | begin a link |
| **<img>** | begin an image |



**An html (HyperText Markup Language) file**

```
<html>
  <head>
     <title>FacultyApplet</title>
  </head>
  <body>
     <p align="center"><B>This</B> is
           an <i>Applet!</i>
     </p>
     <br><br>

     <p><applet archive="AppletClasses.jar"
          code="FacultyApplet.class"
          width=800   height=550>
        </applet>
     </p>

  </body>
</html>
```

tags

| | |
|---|---|
| **<html>** | start an html page |
| **<head>** | start the "heading" |
| **<title>** | the title for the page |
| **<body>** | start the body, content, of the page |
| **<p>** | begin a paragraph |
| **<b>** | begin boldface |
| **<i>** | begin italics |
| **<applet>** | start a Java applet |
| **<br>** | line break (no end tag) |