CSCI1530 Computer Principles and Java Programming

Tutorial 10

Array and Sorting

Contents

- Array
  - Basics
- Sorting
  - Introduction
  - Swap between variables / find minimum
  - Selection sort

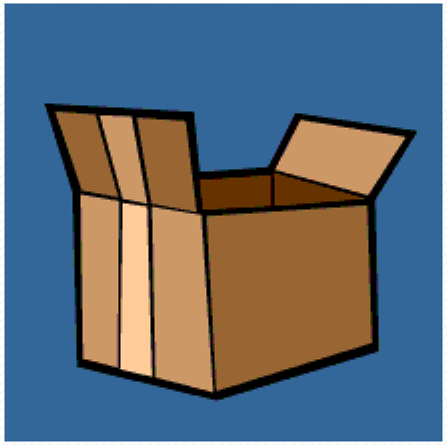
```java
import java.util.*;

class Example {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      int grade1, grade2, grade3;

      System.out.print("Student 1: ");
      grade1 = scanner.nextInt();
      System.out.print("Student 2: ");
      grade2 = scanner.nextInt();
      System.out.print("Student 3: ");
      grade3 = scanner.nextInt();

      System.out.println("Average = " +
                        (grade1 + grade2 + grade3) / 3.0);
   }
}
```

- The program works if there are only three students.
- What if there are 100 students?

Array to the rescue!

Ordinary Variable

Like a box for storing one value



Array

Like a cabinet containing many drawers.

Each drawer stores one value.

We can refer to each drawer as 1st drawer, 2nd drawer, 3rd drawer, etc.

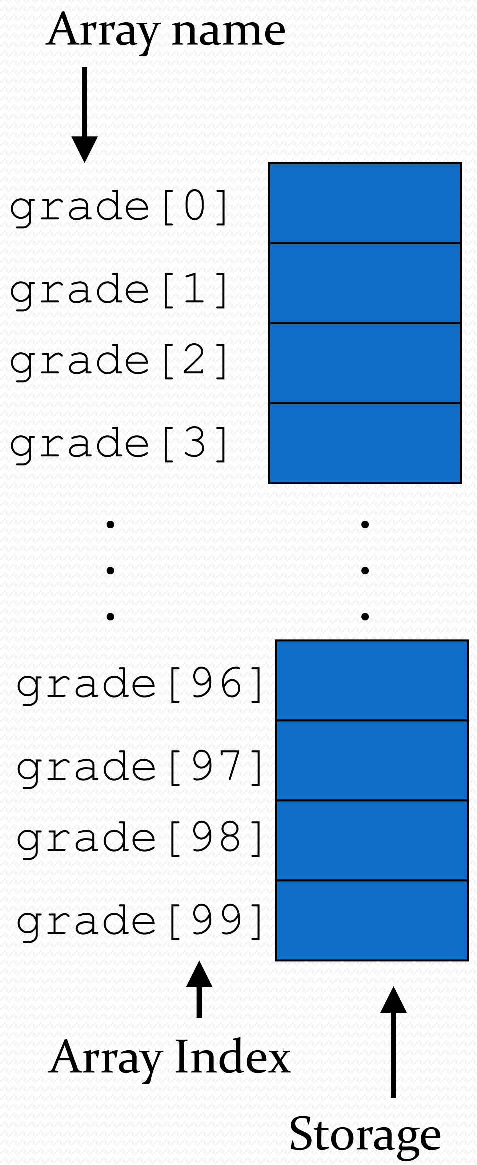
Array

- Stores same type of data
- Array size = number of elements in the array
- Array size remains unchanged throughout program execution

- To refer to an array element

  `arrayname[ index ]`

  - Index always starts from 0
  - Index to last element is (array size – 1)



```java
import java.util.*;

class Example {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      int[] grade = {0, 0, 0};

      for (int i = 0; i < 3; i++) {
         System.out.print("Student " + i + " : ");
         grade[i] = scanner.nextInt();
      }
      System.out.println("Average = " +
               (grade[0] + grade[1] + grade[2]) / 3.0);
   }
}
```

- Array element can be accessed at variable index or numerical index (But the index should be specific and inbounds).

Declaring An Array with Initializers

- Use *initializer list*
  - Items enclosed in braces {}
  - Items in list separated by commas

```
int n[5] = {10,20,30,40,50};
int n[] = {10,20,30,40,50};
//Alternative:
int [] n = {10,20,30,40,50};
```

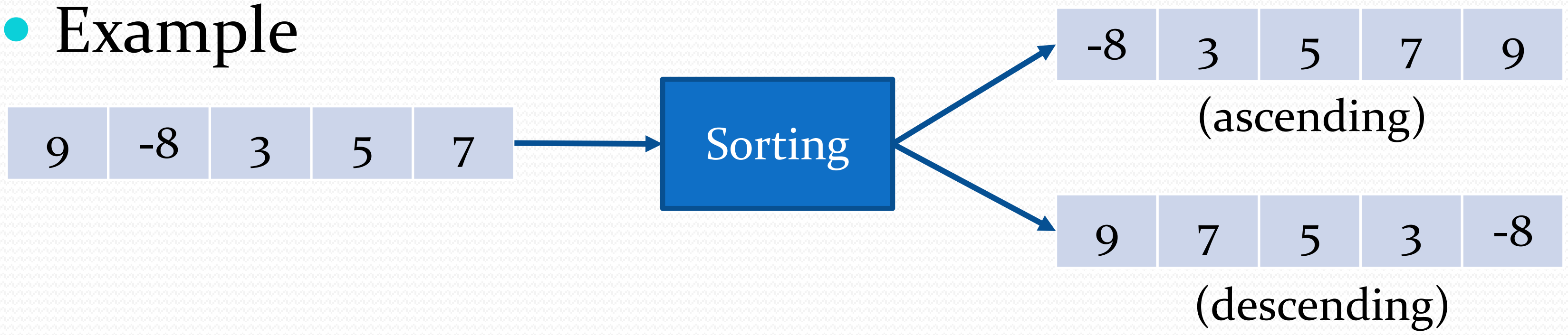
The array length will be automatically set as number of given values

  - Creates a five-element array
  - Index values of 0, 1, 2, 3, 4
- Other examples:
  - String [] str = {"hello","world"};
  - double [] temperature = {23.5,25.4,30.1,19.8};

Sorting

Sorting

- To put a set of data/sequence in order
- Example



- There are many strategies (algorithms) for sorting
- One of the simplest ones is ***Selection Sort***

Sorting algorithms

- Animation
  - http://www.sorting-algorithms.com
- Dance
  - https://www.youtube.com/user/AlgoRythmics

- Which algorithm is the best?
- Pair-wise operation
  - Compare and swap (move)

Swap two numbers

- Given integers `i` and `j`

-
  ```java
  int i = 8, j = 3;

  i = j;          /* i becomes 3 */
  j = i;          /* j also becomes 3 */
  ```
  ✗

```java
int i = 8, j = 3;
i = (j = i);    /* j becomes 8, */
                /* i becomes j, thus 8 also */
```
✗

Create a temp. variable

```
int i = 8, j = 3;
int temp;
temp = j;   /* keep a copy of value of j */
j = i;      /* overwrite j by value of i */
i = temp;   /* restore value of j to i */        ✓
```

- Given us 2 lockers, swap their contents
  - Get the Giraffe out and put it **aside**
  - Move the Elephant to the emptied locker
  - Store the Giraffe into the other emptied one

Sort 2 numbers

- Given 2 integers stored in `i` and `j`

```
int i = 8, j = 3;
if (j < i) {             /* if j is smaller */
    int temp = i;        /* exchange contents of i and j */
    i = j;               /* making the smaller value in i */
    j = temp;
}
System.out.println("Ordered: %d,%d\n", i, j);
```

- How about sorting three, four, or any?
- An array is needed
- To swap the order on the fly

Find the min. of N numbers

```
int N[] = {5, 1, 3};
int i, min;
min = N[0];
for (i = 1; i < 3; i++)   /* note: start with i = 1 */
    if (N[i] < min)
        min = N[i];
```

- At the beginning, let the 1st element as the min.
- Then, compare the element one-by-one with the current min.
    - When current item > min, swap it!

General idea of Selection Sort

- Firstly, we find the 1st min. , then the 2nd min. , … until the last one
- In each round, we put the min. on left hand side
- Without the intervention of the previous min.(s) , we go to next round

- Selection-sort with Gypsy folk dance
  - http://www.youtube.com/watch?v=Ns4TPTC8whw

Example of Selection Sort

```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```

N[0]	N[1]	N[2]	N[3]
4	5	3	1

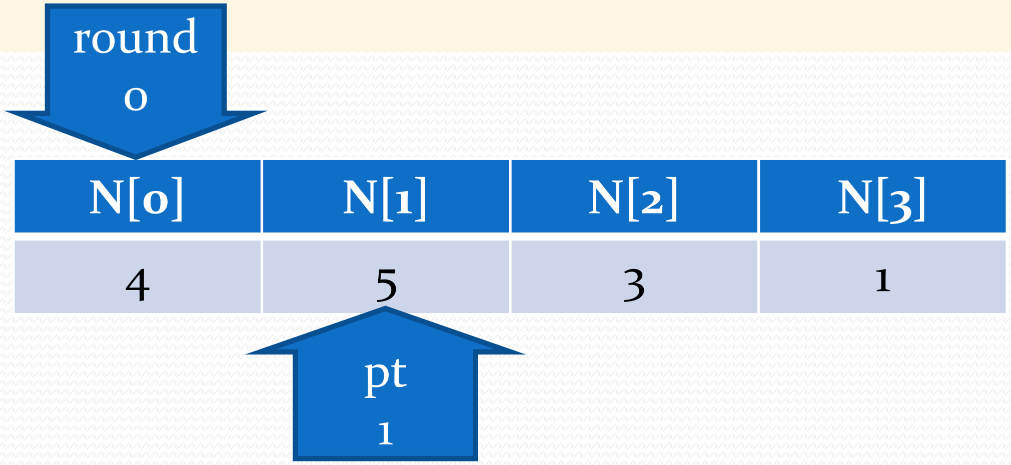
Example of Selection Sort

```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```

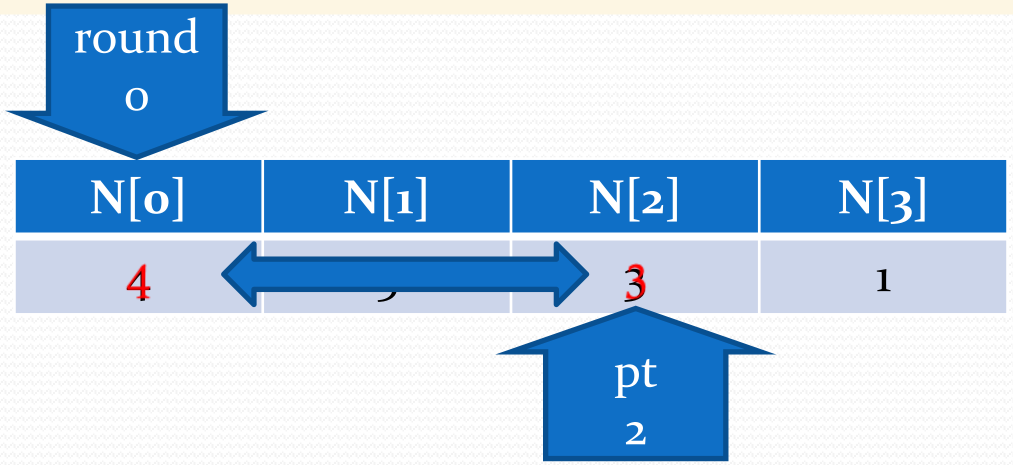
N[0]	N[1]	N[2]	N[3]
4	5	3	1

Example of Selection Sort

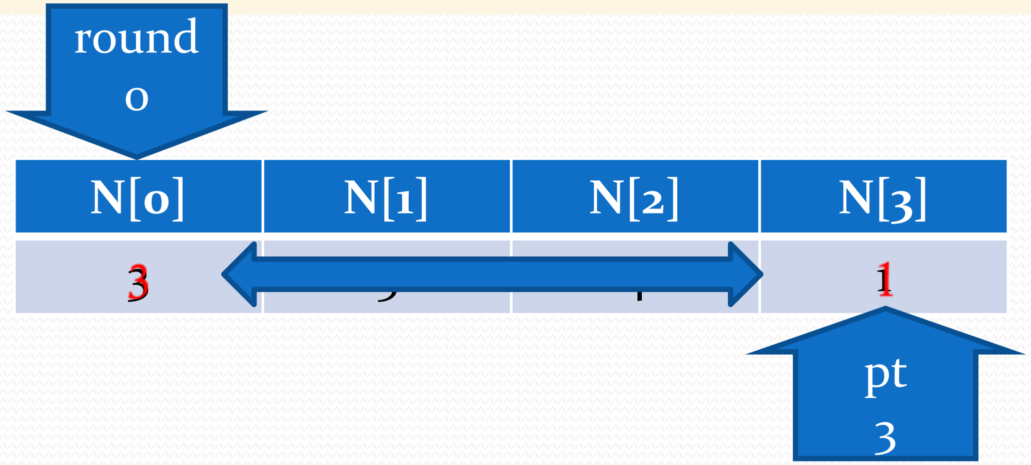
```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```



```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```

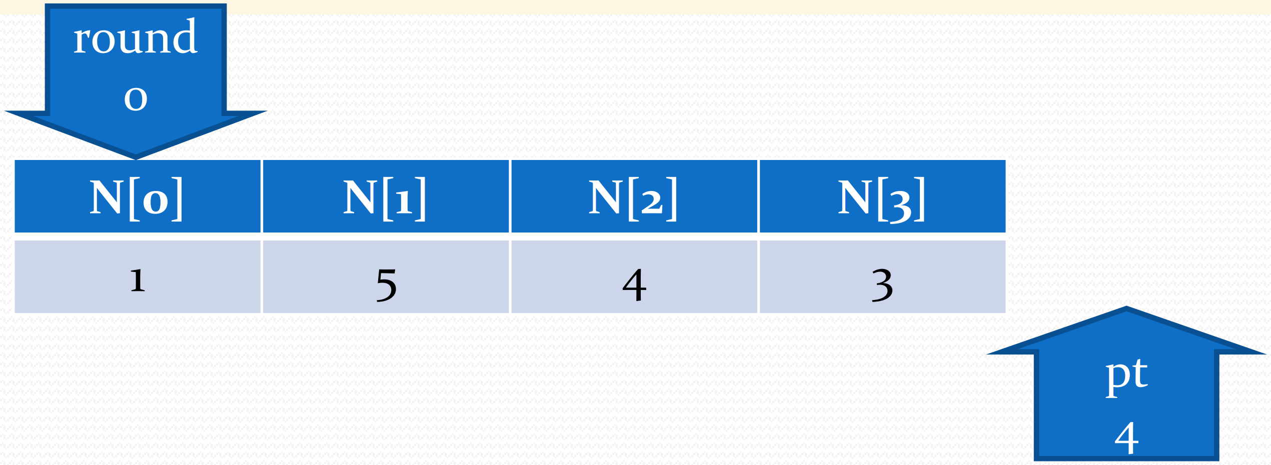


```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```



Example of Selection Sort

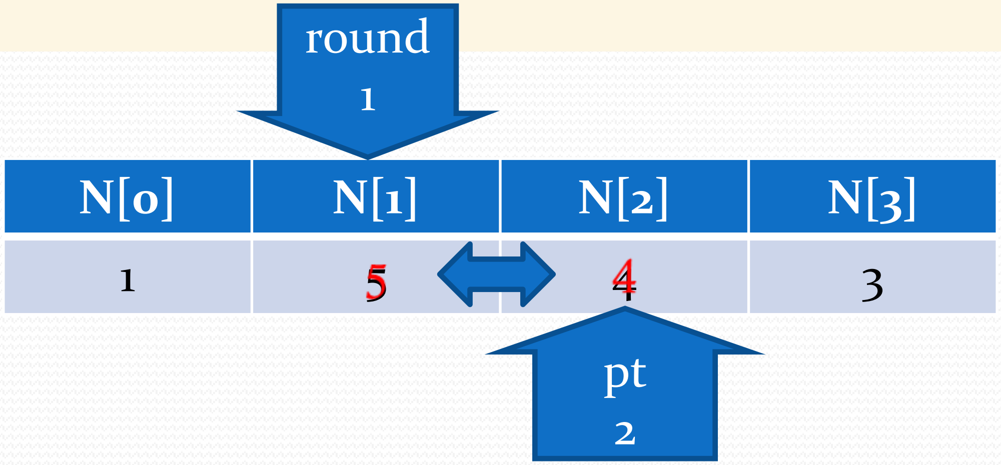
```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```



N[0]	N[1]	N[2]	N[3]
1	5	4	3

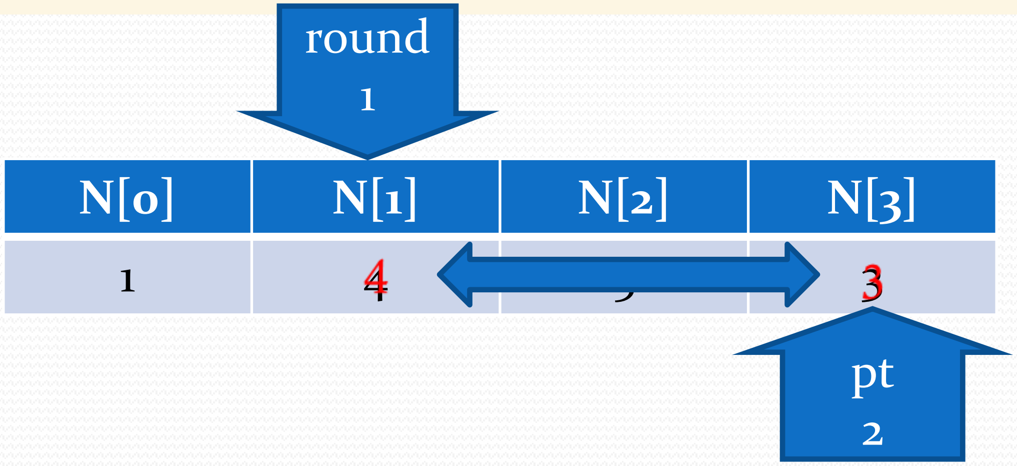
Example of Selection Sort

```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```



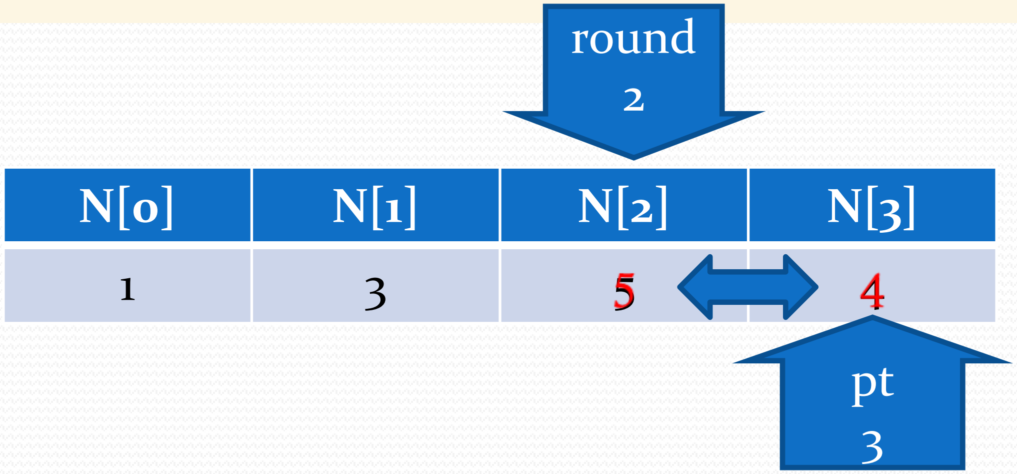
Example of Selection Sort

```java
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```



Example of Selection Sort

```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```



Example of Selection Sort

```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```

Round	N[0]	N[1]	N[2]	N[3]
0	4	5	3	1
1	1	5	4	3
2	1	3	5	4
End	1	3	4	5

Question?