

PROGRAMMING

Lecture 20

Dept. of Computer Engineering
Hanbat National University

Global coordinate data

Temperature data

HTML files

PROBLEM 1: GLOBAL COORINATE DATA

The file, **average-latitude-longitude-countries.csv** contains global coordinate data by nations. **Every line** of the file has the global coordinate data of a nation in the **following format**:

country code, country name, latitude, longitude

where

$-90.0 \leq \text{latitude} \leq 90.0$ and $-180 \leq \text{longitude} \leq 180$.

For example, "KR", "Korea, Republic of", 37, 127.5 for Republic Korea. The **data elements** in a line is **separated** by **commas**. Notice that **some countries** such as Republic of Korea **contains** a **comma** in their **names**.

1-1. Read the file and do the following tasks:

- **Print out** the **global coordinate data** by **countries**
- **Create** a **list** of **tuples** in the following format.

[...(country code, country name, latitude, longitude),...]

The tuples should be sorted in the **alphabetical order** of country codes. You are required to use the recursive version of **merge sort** to do it. The types of data elements are:

country codes and names: str(string)

latitude and longitude: float

How to separate data elements in a line

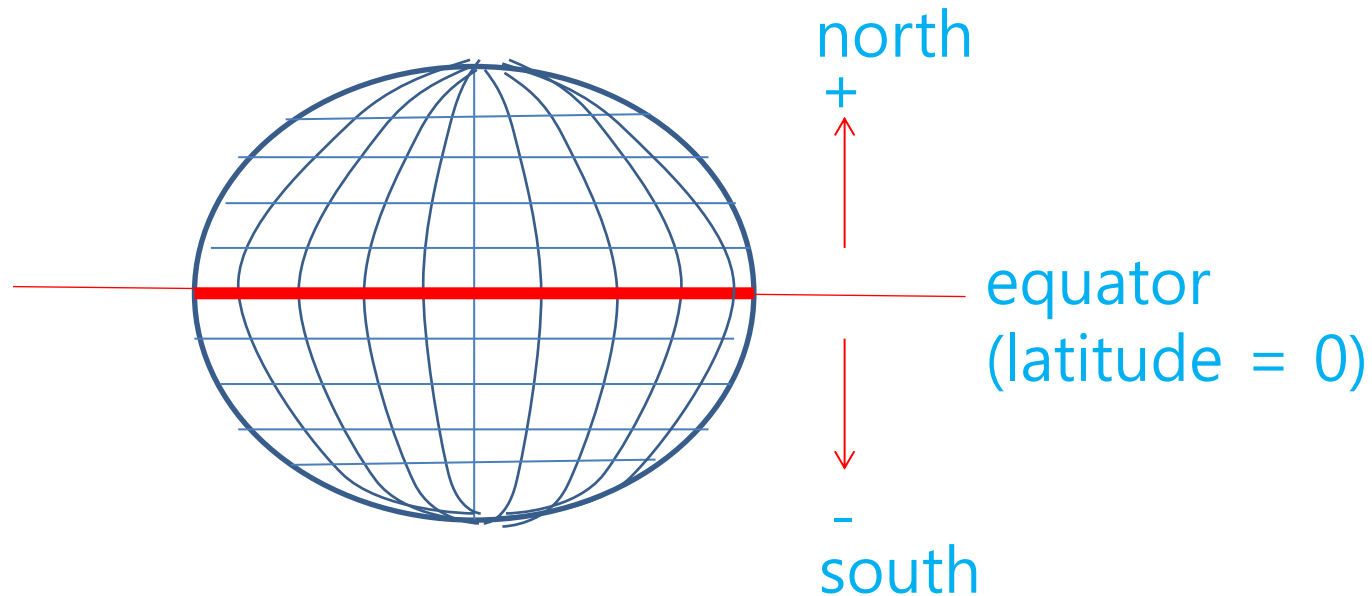
"KR", "Korea, Republic of", 37, 127.5Wn

```
f = open("average-latitude-longitude-countries.csv", "r")
for line in f:
    line = line.strip("\n")
    elements = line.split(",")
    code = elements[0].strip('')
    elements[1] = elements[1].strip('')
    if len(elements) == 4:
        name = elements[1]
        latitude = float(elements[2])
        longitude = float(elements[3])
    else:
        elements[2] = elements[2].strip('')
        name = elements[2] + elements[1]
        latitude = float(elements[3])
        longitude = float(elements[4])
```

How to sort tuples

- Use the recursive version of **merge sort** discussed in a previous lecture
- **Merge** two sorted **lists** using the **country codes** as **comparison keys**

1-2. List out the **names** of **all countries** which are in the south of the equator:



1-3. Suppose that the user provides the country code interactively. Given a **country code**, print the corresponding **country name**. You should use the **recursive version of binary search** to do it.

1-4. Modify your program to use a **dictionary** instead of the list and solve problems 2 and 3.

PROBLEM 2: MONTHLY TEMPERATURE OF ENGLAND

The text file, “**tpmon.txt**” contains temperature information of London for years 1723 ~ 1970. Each line of the file has the monthly average temperatures of a year in the following format:

temp 1 temp 2 temp 3 temp 12

where temp i denotes the temperature of the i -th month of the year. temp i and temp $i+1$ are separated by two space characters (or two blanks). Using this file perform the following two tasks.

2-1. Read the file and print the average summer and winter temperature report as follows:

```
.....  
1737:          5.2/15.6  
1738:          4.6/16.2  
1739:          5.4/15.4  
1740:        -2.2/15.0  
1741:          3.1/16.1  
1742:          2.8/15.8  
1743:          4.5/15.9  
.....
```

Use a **format string** to make the report.
(Continued)

The average summer and winter temperatures are computed as follows:

winter temperature = (January's + February's) / 2

summer temperature = (July's + August's) / 2

2-2. Create the temperature data file (file name: montp.csv) in a csv format in which comma, " , " is used as the separator:

```
1723,1.1,4.4,7.5,8.9,11.7,15.0,15.3,15.6,13.3,11.1,7.5,5.8  
1724,5.6,4.2,4.7,7.2,11.4,15.3,15.0,16.2,14.4,8.6,5.3,3.3  
1725,4.4,3.3,5.0,8.1,10.8,12.2,13.8,13.3,12.8,9.4,6.9,3.9
```

.....

.....

Open the file in **Excel** to confirm it.

PROBLEM 3: EXTRACTING DATA FROM HTML

Open the webpage,

<http://weather.naver.com/rgn/cityWetrCity.nhn?cityRgnCd=CT007023>

and write the information in the webpage in a HTML file, "[./weather.html](#)". Open this file in a web browser to display weather information in the file. In order to create the HTML file, you may use the code in the following slides.

```
import urllib
url = http://weather.naver.com/rgn " +
      "/cityWetrCity.nhn?cityRgnCd=CT007023"
fname = "./weather.html"

dates = []
min_tp = []
max_tp = []
```

def process_webpage():

```
webpage = urllib.urlopen( url )
out = open( fname, "w" )
for line in webpage :
    out.write( line.strip() + "\n" )
webpage.close()
out.close()
```

def main():

```
process_webpage()
print_weather()
```

main()

Extract only the information on date and temperature from the HTML file. Notice that this file is a text file containing this information:

----- A new date starts here. -----

<th scope="row">수요일
(3/05)</th>

<li class="nm">오전 1<em class="dgr">°C

<li class="nm">오후 7<em class="dgr">°C

morning temperature

afternoon temperature

Your task is to print the extracted information as shown below:

5/11 :	12.0 ~ 25.0
5/12 :	11.0 ~ 26.0
5/13 :	13.0 ~ 23.0
5/14 :	15.0 ~ 21.0
5/15 :	13.0 ~ 23.0