Programming: The Next Step

Genome 559: Introduction to Statistical and Computational Genomics

Elhanan Borenstein
A quick review

- **Dictionaries:**
  - key:value pairs
  - a.k.a. hash tables, lookup tables
  - Examples:
    - Word and definition
    - Name and phone number
    - Gene name and score
    - Username and password
  - Dictionaries are useful when you want to look up some data (value) based on a key
  - **Each key can appear only once**

- **Standard I/O**
Note: dictionary and list access times

- Accessing a list by index is very fast!
- Accessing a dictionary by key is very fast!
- Accessing a list by value (e.g. list.index(myVal) or list.count(myVal)) can be SLOW.

by index:

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>val1</td>
</tr>
<tr>
<td>1</td>
<td>val2</td>
</tr>
<tr>
<td>2</td>
<td>val3</td>
</tr>
<tr>
<td>3</td>
<td>val4</td>
</tr>
<tr>
<td>4</td>
<td>val5</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>last_val</td>
</tr>
</tbody>
</table>

(index points directly to position in memory)

by value:

<table>
<thead>
<tr>
<th>Value</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>val1</td>
<td>0</td>
</tr>
<tr>
<td>val2</td>
<td>1</td>
</tr>
<tr>
<td>val3</td>
<td>2</td>
</tr>
<tr>
<td>val4</td>
<td>3</td>
</tr>
<tr>
<td>val5</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>last_val</td>
<td>max</td>
</tr>
</tbody>
</table>
import sys
matrixFile = open(sys.argv[1], "r")
matrix = []  # initialize empty matrix
line = matrixFile.readline().strip()  # read first line stripped
while len(line) > 0:
    fields = line.split("\t")  # split line on tabs, giving a list of strings
    intList = []  # create an int list to fill
    for field in fields:
        intList.append(int(field))  # append the int value of field to intList
    matrix.append(intList)  # after intList is filled, append it to matrix
    line = matrixFile.readline().strip()  # read next line and repeat loop
matrixFile.close()

for row in matrix:
    for val in row:
        print val,
    print ""

... and think how much you've learned!

4 weeks ago, this would have been gibberish:
In theory, what you know so far allows you to solve any computational task ("universality")

So ... why don’t we stop here?
most real-life tasks will be (very) painful to solve using only what you know so far ...
What are we missing?
What are we missing?

- A way to generalized procedures ...
- A way to store and handle complex data ...
- A way to organize our code ...
- Better design and coding practices ...