CSE 12:
Basic data structures and object-oriented design

Jacob Whitehill
jake@mplab.ucsd.edu

Lecture One
2 July 2012
Administrivia.
Course structure

• Lectures:
  • M,T,W,Th
    2:00p - 3:20p
    Center Hall 105

• Discussion sections (3/week):
  • TBA
Grading

- 4 or 5 programming assignments (45%)
- 1 midterm examination (20%)
- 1 final examination (30%)
- In-class unannounced quizzes (yay!) (5%)
Teaching staff

• Lecturer: Me

• Teaching assistant: Ruixin Yang

• Tutors/graders:
  • Kerwin Azares
  • Yuta Murinaga
  • Kimberly Chu
  • Susan Liu
  • Wilson Guo
  • Dustin Yu
Course website

• ieng6.ucsd.edu/~cs12u

• JAKE -- open this page
Moodle forum

• http://csemoodle.ucsd.edu

• Appropriate contributions:
  
  • Questions about programming projects, data structures, or anything else in computer science.

  • Answers to the above.

  • Suggestions for topics you want to hear about during lecture and/or discussion section.
Discussion section

• Go to CSE 12 Moodle web forum, follow the Doodle link, and list your availability.
Before we begin

• The word “data” is technically a plural.

• I will usually say “data are...” or “they are” instead of “data is” or “it is”.

• (Unless it sounds overly pretentious.)
Storing information in a river of 1’s and 0’s.
Data structures

• What are data?

• Why do we need to structure them?
Data structures

• What are data?
  • Data are just a sequence of 1’s and 0’s.

• Why do we need to structure them?
Data structures

• What are data?
  • Data are just a sequence of 1’s and 0’s.

• Why do we need to structure them?
  • Because data are just a sequence of 1’s and 0’s.
Data structures

- Without knowing the structure of the data, we have no idea what it means.
- The following bitstring could contain:
Data structures

• Without knowing the **structure** of the data, we have no idea what it means.

• The following bitstring could contain:
  • An image of UCSD campus
Data structures

• Without knowing the **structure** of the data, we have no idea what it means.

• The following bitstring could contain:
  
  • An image of UCSD campus
  
  • A movie file for *Bambi*
Data structures

• Without knowing the **structure** of the data, we have no idea what it means.

• The following bitstring could contain:
  • An image of UCSD campus
  • A movie file for *Bambi*
  • A recipe for lasagna

1100010100010111010110101011010010000111111001110111001010111110001010101101011001000101111010111...
Consider the following request:

• Dear you,

    Please email me your phone number.

Thanks,
Someone else
Transmitting your phone #

• To transmit your phone number by email, the 10 digits must be converted into a binary sequence of 1’s and 0’s.

• That’s all you (ever) have to work with.
Converting your phone # into binary data

- Phone numbers in USA have 10 decimal digits, e.g., (858) 822-5241.

- How do we convert this phone number into a bitstring so that the receiver can recover the original message?

  - E.g., the receiver has to decode somehow what phone number is stored in: 1000010110001000001000100101001001000001
Converting your phone # into binary data

- Phone numbers in USA have 10 decimal digits, e.g., (858) 822-5241.

- How do we convert this phone number into a bitstring so that the receiver can recover the original message?

- One possible strategy:
  - Encode each digit separately as a 4-bit string (40 bits total).
Encode each digit

• Examples:

• To encode digit 8 using 4 bits, we write 1000.

• To encode digit 1 using 4 bits, we write 0001.

• Given the binary codes for each decimal digit, we concatenate all the codes together (in order), e.g.,

   8 5 8 8 2 2 5 2 4 1

   1000
Encode each digit

- Examples:
  - To encode digit 8 using 4 bits, we write $1000$.
  - To encode digit 1 using 4 bits, we write $0001$.
- Given the binary codes for each decimal digit, we concatenate all the codes together (in order), e.g.,

  8  5  8  8  2  2  5  2  4  1

  1000 0101
Encode each digit

- **Examples:**
  - To encode digit 8 using 4 bits, we write 1000.
  - To encode digit 1 using 4 bits, we write 0001.

- Given the binary codes for each decimal digit, we concatenate all the codes together (in order), e.g.,

  8  5  8  8  2  2  5  2  4  1
  1000 0101 1000
Encode each digit

• Examples:
  • To encode digit 8 using 4 bits, we write 1000.
  • To encode digit 1 using 4 bits, we write 0001.

• Given the binary codes for each decimal digit, we concatenate all the codes together (in order), e.g.,
  \[
  \begin{array}{cccccccccc}
  8 & 5 & 8 & 8 & 2 & 2 & 5 & 2 & 4 & 1 \\
  1000 & 0101 & 1000 & 1000 & 0010 & 0010 & 0101 & 0010 & 0100 & 0001 \\
  \end{array}
  \]
Encode each digit

- Examples:
  - To encode digit 8 using 4 bits, we write 1000.
  - To encode digit 1 using 4 bits, we write 0001.
  - Given the binary codes for each decimal digit, we concatenate all the codes together (in order), e.g.,

\[1000010110001000001001001000001\]
Transmitting your phone #

- We then send this bit sequence over the network to Someone else.

Your computer

100010110...

Network

Someone else’s computer
Transmitting your phone #

- We then send this bit sequence over the network to Someone else.
Transmitting your phone #

- We then send this bit sequence over the network to Someone else.
Transmitting your phone #

- We then send this bit sequence over the network to Someone else.
Transmitting your phone #

- We then send this bit sequence over the network to Someone else.

Your computer

Someone else’s computer

1000010110...
Decoding your phone #

- Someone else then has to **decode** the bit sequence you sent:

  `100001011000100000100100101001001000001`

- How?
Decoding your phone #

• Someone else then has to **decode** the bit sequence you sent:

```
1000010110001000001000100101001001000001
```

• How?

• The rule that “every 4 bits = 1 digit” means that Someone else can divide the bitstring into **slots**.

```
1000 0101 1000 1000 0010 0010 0101 0010 0100 0001
```
Decoding your phone #

• Someone else then has to decode the bit sequence you sent:

1000101100010000010010001001000001

• How?

• The rule that “every 4 bits = 1 digit” means that Someone else can divide the bitstring into slots.

• Dividing data into slots is a useful technique that will recur throughout the course.
Decoding your phone #

• Consider the binary code for decimal digit 1:
  • 0001

• What if we try to “save space” by omitting the leading zeros? Aren’t leading zeros useless anyhow?
Decoding your phone #

• Consider the binary code for decimal digit 1:
  • 0001

• What if we try to “save space” by omitting the leading zeros? Aren’t leading zeros useless anyhow?

• Unfortunately not -- they help to structure the data by ensuring a constant length of each digit.
Decoding your phone #

• Consider the number 515...:
  • 0101  0001  0101 ...

• Without the leading 0’s, we have:
  • 101  1  101 ...

• The problem is that Someone else doesn’t see the spaces -- all they see is 1011101.

• No way to infer where each digit starts/ends.

• We need to structure the data by making each decimal digit have the same length.
A slightly more ambitious task...
How would you handle this?

• Dear Google,

Please send me all of your Google Earth data.

Thanks,
Some Other Company
Data transmission

- To handle such a request, we could either:
  - Transmit the data over a network using a very long sequence of 1’s and 0’s.
  - Write the entire Google Earth database to a large number of hard disks (containing 1’s and 0’s) of high capacity.
- Ultimately, we need to encode a huge amount of information as 1’s and 0’s.
I’s and 0’s

• What we would like to do is send separate “groups” of bits for different parts of the data:

Images of Australia

1101010000110011010
1001100101011101010
0110100011001000101
1101000101000101001
0110000111010110100
101010011001001011
1010001...

Images of Europe

0000001000000110100
0011100001110110011
1001110001101100000
1001011001010110001
0110101011111010010
00111...

1011011010
0000001111
0010000001
1100101110
... Marker of Eiffel Tower

1001001110
0010101110
11011...

Marker of Mt. Everest
I’s and 0’s

• In the real world, we must unfortunately encode everything in a single stream of I’s and 0’s.

• We must somehow structure our data (I’s and 0’s) to allow meaningful information to be extracted.

• First, how many bits are we dealing with?
How much data is there in Google Earth?

- Satellite imagery: 510,072,000 km²
  * $1000^2$ pixels/km² * 3 bytes/pixel
  * 8 bits/byte =
  12,241,728,000,000,000 bits
- (About 12 quintillion bits)
Satellite imagery

- How might we store the satellite imagery?
  1. Convert image of 3-D spherical surface to 2-D image.

3-D sphere  ➔  Mercator projection  ➔  2-D image
Satellite imagery

• What is an image in terms of 1’s and 0’s?

  1. Image is a 2-D grid of pixels.

[Diagrams of a world map and a 2-D grid of pixels]
Satellite imagery

• What is an image in terms of 1’s and 0’s?

2. Each pixel consists of red, blue, and green color channels.
Each color channel is between 0-255.
Satellite imagery

- What is an image in terms of 1’s and 0’s?

2. We can represent a 2-D image as a \(3N \times M\) array of numbers.

\[
\begin{array}{cccccccc}
255 & 255 & 255 & 255 & 255 & 255 & \ldots \\
255 & 14 & 23 & 24 & 24 & 23 & \\
255 & 17 & 25 & 27 & 25 & 24 & \\
255 & 255 & 255 & 255 & 255 & 255 & \ldots \\
255 & 255 & 255 & 255 & 64 & 64 & \\
255 & 53 & 61 & 63 & 64 & 64 & \\
\ldots & & & & & & \ldots \\
\end{array}
\]

\((R,G,B)\) for one pixel
Satellite imagery

• What is an image in terms of 1’s and 0’s?

3. We can concatenate the $3N \times M$ array elements into one large vector of numbers.

2-D array

I-D vector of numbers

N rows

M columns

...
Satellite imagery

- What is an image in terms of 1’s and 0’s?

4. We convert each element of the vector (0-255) to binary representation.

I-D vector of integers  
I-D binary vector

Done!
Storing the markers

- Each marker consists of:
  - Location (latitude & longitude)
  - Name

- Example:
  - **Rock Bottom Brewery, La Jolla:**
    - 32 deg 52’ S latitude,
    - 117 deg 14’ W longitude
Storing the markers

- We can encode each marker in the following way:

```
0001011110110101
1001110111010011
1101010001011111
1011100100110011
0110100010010011
```

<table>
<thead>
<tr>
<th>Latitude minutes</th>
<th>Latitude seconds</th>
<th>Longitude minutes</th>
<th>Longitude seconds</th>
<th>Marker name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Google Earth data: One huge binary sequence

- We concatenate the satellite imagery and markers into one huge binary sequence (serialization).

Satellite image

```
11111110011110011100001011111000011111010101001010101010
0111001100001100000111110010110111011000011110001
1010001011001011111101101110000111111101000110111
11101000101101111010100011100011001000...
```

Markers

- The serialized data can then be easily:
  - Loaded into memory.
  - Written to disk/DVD.
  - Transmitted over a network.
Google Earth data:
One huge binary sequence

- After serializing the Google Earth data, we can send it to Some Other Company.

Google -> Send data -> Some Other Company

...011010001011...
Google Earth data: One huge binary sequence

- After serializing the Google Earth data, we can send it to Some Other Company.
- But -- our encoding is still problematic...

Send data 
...011010001011...

Google  --- Send data -----> Some Other Company
Some other company: How do I parse the 0’s and 1’s?

- How many bits long is the satellite imagery?

<start>
1111110011110011110000101111100001111101010010101010
01110011000001110000111110010110111011000001111001
1010001011100101111111011011000011111101000110111
111010001011011110101000111001011001000...
@end>

- It is necessary to give Some Other Company information on how to “jump” to the markers.
Extracting satellite image data

• How can we let Some Other Company know how long the satellite image data subsequence is?
  • Encode M and N as integers just before the image data?
  • Encode M and N as integers just after the image data?
Extracting satellite image data

- Encode $M$ and $N$ as integers just before the image data?
- $M \times N$ pixels, 3 colors each, 8 bits for each color channel and pixel = $3 \times M \times N \times 8$

Satellite image
$(3 \times M \times N \times 8 \text{ bits})$
Extracting satellite image data

- Encode M and N as integers just **before**
  the image data?

- M*N pixels, 3 colors each, 8 bits for each
  color channel and pixel = 3*M*N*8

M and N
allows us to
“jump” to
the marker
data.

<table>
<thead>
<tr>
<th>M (320)</th>
<th>N (200)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000001010000000000001100100011111100111001</td>
<td></td>
</tr>
<tr>
<td>1100001011111000011111010100101010100111001</td>
<td></td>
</tr>
<tr>
<td>00011000001111100101101101100001111000110100</td>
<td></td>
</tr>
<tr>
<td>010100101111000011111010100101010010101101...</td>
<td></td>
</tr>
<tr>
<td>110010111111101101110000111111010001101111110</td>
<td></td>
</tr>
<tr>
<td>100010110111101010001110001011001001000...</td>
<td></td>
</tr>
</tbody>
</table>

*Satellite*  
*image*  
*Marker*  
data

Monday, July 2, 12
Extracting satellite image data

- Encode $M$ and $N$ as integers just after the image data.

Satellite image
(3*$M$*$N$*8 bits)

$\text{<start>}$
\[
\ldots
11111110011100111000101111100011111010100101010100111001100000110000011111001011011101100001111000110100010100000010100000001010000000000001100
1000\ldots
\text{<end>}

$M$ (320)  
$N$ (200)
Extracting satellite image data

- Encode M and N as integers just \textbf{after} the image data? \textbf{No}

Problem: where in the bitstring are M and N stored?

Satellite image (3*M*N*8 bits)?
Extracting marker data

• Assume:
  • 1st marker starts immediately after satellite data.
  • 2nd marker starts immediately after 1st, etc.
• Similarly to how we specified how big the satellite image is, we must also specify how long each marker is.

  • <start>
    
    ...  
    111111001110011100001011111001111101010001010
    1010011100110000110000011111001011011101100000
    111100011010001011100101111111101101110000111111
    101000110111111010001011110101000111000101110
    01000...
    <end>
Extracting marker data

• At beginning of binary sequence for marker each, we encode its length.

- <start> Marker I length Marker I
  ... 111111001110001101111000011111010100101010
  10011100110000111000011111001110111011100001111
  00011010010101000000110010111111011011100001111
  111010001101111101000101110110001100100011010
  01000... <end>
Extracting Google Earth data

- We can now extract both the satellite image data and the markers.
- How might this look in Java code...?
Data structures and object-orientation.
Google Earth data extraction in Java

- We can use the `Image` class for the satellite image.
- Let’s assume there’s some pre-existing `Location` class to represent latitude+longitude.
- Let’s create a `Marker` class:
Google Earth data extraction in Java

• We can use the `Image` class for the satellite image.

• Let’s assume there’s some pre-existing `Location` class to represent latitude+longitude.

• Let’s create a `Marker` class:

```java
class Marker {
   private String _name;
   private Location _location;
}
```
Google Earth data extraction in Java

• To extract satellite image and markers from the bit sequence, let’s define 2 “pseudo-Java” methods:

• // Should be called at beginning of entire Google Earth bit sequence.
public Image extractSatelliteImage (bit[] sequence)
{ ... }

• // Should be called on the bit sequence just after the satellite data.
public Marker[] extractMarkers (bit[] sequence)
{ ... }

* Type “bit” doesn’t actually exist in Java.
Google Earth data extraction in Java

- Danger 1...
- Danger 2...
Google Earth data extraction in Java

• **Danger 1 -- wrong bits:** the caller calls a method on the wrong part of the Google Earth data bit sequence.

• **Danger 2 -- mismatched image/markers:** if there are multiple planets (Google Earth, Google Mars, etc.), then the caller might mismatch the set of markers with the wrong planet.
In come the objects...

• One of the purposes of objects in Java is to prevent these problems from occurring.

• Objects **encapsulate** related pieces of data.

• Example: define a class `GooglePlanet`. 
class GoogleEarth

• class GooglePlanet {
    Image _satelliteImage;
    Marker[] _markers;

    // Should start at the beginning of entire
    // Google Planet bit sequence.
    GooglePlanet (bit[] sequence) { ... }
}

• Now, the constructor of GooglePlanet
  handles the initialization.

• Fewer opportunities for caller to mess up --
  only one bit sequence to pass in.
class GoogleEarth

• class GooglePlanet {
    Image _satelliteImage;
    Marker[] _markers;

    // Should start at the beginning of entire
    // Google Planet bit sequence.
    GooglePlanet (bit[] sequence) { ... }
}

• Also, the satellite image and markers are
  eternally coupled (how romantic) -- there is no
  danger of mismatching markers and images.
Object-orientation and data structures

• These are two benefits of data encapsulation. (There are others.)

• Data encapsulation is a benefit of object-orientation.

• Other benefits include:
  • Polymorphism
  • Abstraction
  • (More on these later in the course...)
Time complexity and space complexity.
What do we use class GoogleEarthData for?

- How is the Google Earth data used in practice? Common use case:
  - User is navigating somewhere on Earth, and wants to fetch a list of markers nearby (e.g., fish taco restaurants).
Finding local markers

• To implement this “query” functionality, let’s add a method to class Marker:

• (For simplicity, a marker is either “close to” the user’s location, or “not close to” her/him.)

```java
class Marker {
    ...
    public boolean isCloseTo (Location location) {
        ...
    }
}
```
Finding local markers

• We also add a method to GooglePlanet:

```java
class GooglePlanet {
    Image _satelliteImage;
    Marker[] _markers;

    public Marker[] findLocalMarkers (Location location) {
        ...
    }
}
```

• How is this method implemented?
Unannounced quiz 0
Finding local markers

• Algorithm:

Create empty list localMarkers
For each Marker i in _markers:
    If _markers[i].isCloseTo(location):
        Add _markers[i] to localMarkers
Return localMarkers
Time cost

- If there are 10,000,000 markers in _markers, how many times will the loop iterate?

- What if location is not close to any of the markers in _markers?

- Problem -- we have to call isCloseTo() on markers that are very far away from location.
Time cost

• Implementing _markers as a simple array causes findLocalMarkers to run fairly slowly (“linear time” in this case).

• The running time of findLocalMarkers can be estimated from the time complexity of that method.

• The time complexity of an algorithm depends on the data structures it uses.
Finding local markers more quickly

- It would be nice to speed up the search for local markers.
- Simple approach: divide the markers into regions.
- When looking for local markers, we search only within our local region.
Finding local markers more quickly

class GooglePlanet {
  Image _satelliteImage;
  Marker[] _region1, _region2, ..., _region12;
  ...
}
Finding local markers more quickly

• New algorithm for finding local markers:

Create empty list localMarkers
Determine which localRegion contains location
For each Marker i in localRegion:
  If localRegion[i].isCloseTo(location):
    Add localRegion[i] to localMarkers
Return localMarkers
Finding local markers more quickly

- If there are 12 regions, then this algorithm will run about 12x faster than our first one.

- **Time cost** has been reduced.

- BUT -- there is a penalty.

- Instead of just one `Marker[]`:
  ```c
  Marker[] _markers;
  ```
  we now have 12 `Marker[]`'s:
  ```c
  Marker[] _region1, _region2, ..., _region12;
  ```
Array overhead

- Each array of type `Marker[]` incurs some overhead.
  - In Java, the length of an array is stored in its `length` field. This takes up space!

- So...we have decreased the **time cost** at the expense of increasing **space cost**.

- There is an inherent tension between minimizing time cost and minimizing space cost.

- The space cost of a data structure can be estimated from its **space complexity**.
Finding local markers more quickly

- Our “grid” of local regions is still not great in terms of time cost.
- A **tree** data structure could yield much better performance (more later in the course...).
“Code complexity”

• Sometimes, it may be reasonable to sacrifice some time/space costs to make the code simpler.

• Especially on small amounts of data, an “easy to implement” data structure may often be the best solution.
Choosing the right data structure.
Choosing the right data structure

• When writing a program, very often you will be solving the same kinds of problems over and over again:
  • How do I store a collection of addresses?
  • How do I sort these numbers?
  • How can I find the largest object quickly?
  • How can I fetch a person’s profile picture from a dataset quickly given just her name?
Choosing the right data structure

• Rather than having to rediscover the solution every time, you should learn how the fundamental data structures of computer science work.

• Data structures covered in this course:
  • List
  • Stack
  • Queue
  • Heap
  • Tree
  • Hash table
  • Graph
The rest of this course.
• In this course you will study the properties of and practice implementing the data structures listed above (list, stack, queue, heap, tree, hash table, graph).
• **Question:** “Why should I spend time implementing a data structure that has been implemented literally millions of times before, when superbly written, highly efficient, thoroughly tested, standardized library versions exist for free?”
CSE 12

- My answer:
  - Once you thoroughly understand the basic data structures, by all means use library code.
  - BUT: There is no better way of gaining a thorough understanding than having to implement those structures yourself.
Programming Project #1

- In your zeroth programming project (P0) you will write a Hello Whirled program.
- In your first programming project (P1) you will implement a doubly-linked list.
- As the first part of P1, you will test your linked list implementation.
Getting help

- If you need help on the programming project, you can come to:
  - Me during office hours or in the lab.
  - The TA during discussion section, his office hours, or in the lab.
  - The tutors in the lab.
  - The web forum (csemoodle).
  - Each programming project will be allocated its own thread.
Getting help

• You may also get help from:
  • Your peers.
  • This is natural.
  • This is beneficial.
  • This is slightly dangerous...
Obtaining help from peers

- It is ok to talk to your peers about CSE 12 programming assignments without writing any notes on paper or on the computer.
- Equivalent to having a phone call.
- It is ok to discuss a programming assignment while using paper and pencil as a visual aid.
- **BUT:** you must destroy these notes before returning to your computer.
- You may not look at someone else’s code on any computer screen.
Obtaining help from the Internet

- Feel free to consult general texts on data structures on the Internet:
  - Online textbooks
  - Wikipedia
- You are *not* permitted to download anyone’s source code to complete an assignment.
Enforcement

• We will be using automatic code comparison programs to identify copied code.

• In a previous course I taught, I caught one student cheating; he failed the course and his graduation was delayed by 1 year :-(.
Participation in class

• Please ask questions during class if you are curious about or do not understand something.

• It is not a bother to answer questions.

• Answering questions is my job.

• Answering “stupid” questions is my job.

• Every student (and instructor) sometimes makes mistakes.

• Please show respect to classmates (and me) at all times.
End