Recursion

Admin

- Quiz paper
- Unfortunately recursion is not in this book.
- Last outcome that we haven’t done at all

Solving problems with a smaller version of problem

- In Math
  - many problems are solved using smaller versions of the same problem
  - Fibonacci numbers
  - 1 1 2 3 5 8
- In Philosophy
  - inductive proofs
  - show base case is true
  - show that each later case follows from simple step and earlier case

In Computer Science

- We can use the same inductive technique in our programming
  - Recursion.
  -
- Recursion:
  - Solving a problem by using the solution to a simpler version of the problem and a small additional bit of work

Recursive Definitions

- Have you had a teacher tell you that you can’t use a word in its own definition? This is a *circular* definition.
- In mathematics, recursion is frequently used. The most common example is the factorial:
- For example, 5! = 5(4)(3)(2)(1), or 5! = 5(4!)
- Use board? since it often looks ugly cross platform on slide.

Recursive Definitions

- Recursive definitions aren’t circular because they eventually have a base case, that can be instantly computed without further work
- Every recursive solution has two parts
  - A base case which can be instantly computed
  - A recursive case, which does a little bit of work, and then calls the same function to do a simpler (closer to the base case) version of the same problem.

Let’s try

- Try factorial with recursion
-

a nice problem for recursion.

- Problem: test whether a sentence is a palindrome
  - Palindrome: a string that is equal to itself when you reverse all characters
    - A man, a plan, a canal–Panama!
    - Go hang a salami, I'm a lasagna hog
    - Madam, I'm Adam
  - how would you design a recursive solution to this problem?

Sample code minus solution

- Let's assume all of the non-alphanumeric characters have been removed – let's build a recursive function to check if that string_to_check is a palindrome

```
def is_palindrome(string_to_check):

    #fill in here
```

So what do we do first?

- What are the two parts of a recursive solution?

So what do we do first?

- What are the two parts of a recursive solution?
- And which of those comes first?

So what do we do first?

- What are the two parts of a recursive solution?
- And which of those comes first?

- Base case
  - Always has to be first
- Recursive case

So what is a good base case?

- What is a good base case for a recursive palindrome checker?
  - What string can you determine immediately is (or is not) a palindrome

Recursive Case

- Once we have a base case(s) we need our recursive case(s)
- Do a tiny bit of work, then make a recursive call to do the rest
  - Recursive call needs to be in some way closer to the base case.

- Lets work through the example in pycharm

Recursion and Iteration

- Recall recursive factorial and iterative factorial
  - Anything you do with recursion you can do with iteration/looping
  - And we saw that
- But sometimes one is better than the other.

Lets try a recursive Fibonacci

- Remember
- fib(1) → 1
- fib(2) → 1
- fib(n) → fib(n-1)+fib(n-2)

- Lets implement that with a recursive solution.
- And run it

What happened?

- What happened?
  - Recursive factorial worked fine
  - But recursive fibonacci fell down quickly

- Lets look at it in the board

What happened?

- What happened?
  - Recursive factorial worked fine
  - But recursive fibonacci fell down quickly

- Lets look at it in the board
- Moral of the story if more than one recursive call is working on the same data, then it is going to be bad.

Searching

- Searching through data:
  - Looking for a particular value in a collections
- Search through a list of student records for one with your banner id so you can register

Python built in search

- To check to see if a value is in a list or not (review from earlier in the semester)
  - If 'John' in names:
    - #Hooray I'm here
  - Use keyword in to check to see if some value is in a list.
  - Returns true if value is in list, returns false otherwise
  - Does a linear search of list. (show on board)
- The other way: If we know it is in the list and want to know where
  - names.index('John')
  - Use index method on list object.

Searching for an object

- We have as list of students and a bannerId
  - We want to find the student record in the list with just the bannerID
  - The student record has the rest of the interesting stuff like GPA and student name and more
  - So how do we (humans) search?

Searching for an object

- We have as list of students and a bannerId
  - We want to find the student record in the list with just the bannerID
  - The student record has the rest of the interesting stuff like GPA and student name and more
  - So how do we (humans) search?
    - So you do it by flipping through and looking.
    - Computer can't do that
      - Without any assumptions
    - Computer must begin from beginning and look through data to find it
    - (this is what in and index do)

Assumptions about data

- In the last slide I said without any assumptions about data
  - But what if we can make assumptions about the data?
  - How can we make it easier to search?

Assumptions about data

- In the last slide I said without any assumptions about data
  - But what if we can make assumptions about the data?
  - How can we make it easier to search?
  - If the data is ordered (sorted) then we don't have to look at every piece of data
    - We know by looking as one item, that lots of the data is either more than what we are looking for or less than what we are looking for

Sorting

- Many times you will have to sort a list
  - Lots of algorithms only work when data is sorted
  - In some CS courses we will do that ourselves
    - Need to know how a car works before we can design one
  - In most of the discipline we will use the built in sorting
  - Warning to those of you in MIS

So lets sort

- Python has two mechanisms for producing sorted lists
- List class has a sort method
  - aList = [1,4,2,3]
  - aList.sort() # list is now [1,2,3,4]
- Also sorted function
  - Returns a sorted copy of the list leaving the original list unchanged

Sorting something useful

- Sorting a list of integers is not interesting
  - What if we want to sort that list of student objects?
  - How will that work?
  - Again – how will python know what to sort on? What does it mean for one row to be larger than another?

Find the largest

- First – a philosophical question what does it mean to be the largest?
    - With numbers that is easy right?
        - Number furthest in the positive direction on the mythical number line
    - What about strings?
        - The one with the most characters
  - What about student objects?
      - Lets look at a student object on the board

Find the largest

- First – a philosophical question what does it mean to be the largest?
    - With numbers that is easy right?
        - Number furthest in the positive direction on the mythical number line
    - What about strings?
        - The one with the most characters
  - What about student objects?
    - Lets look at a student object on the board
    - Sort based on GPA? Student ID, alphabetical by student name?

- Sorting a list of integers is not interesting
  - What if we want to sort that list of Student Objects?
  - How will that work?
  - Again – how will python know what to sort on? What does it mean for one student to be larger than another?
    - In the end, we have to tell python how we want it sorted.

Defining a sort key

- If we can come up with a numeric sortable representation of each object
  - python can order the objects
  - Define get_key
    - Function should take an object (a student for us) and return a number for the student object to be sorted on
    - What should we use for our number for student objects?

Defining a sort key

- If we can come up with a numeric sortable representation of each object
  - python can order the objects
  - Define get_key
    - Function should take an object (a student for us) and return a number for the student object to be sorted on
    - What should we use for our number for student objects?

Defining get_key

- How will we define get_key?
  - Make sure param and return value is right
  - Remember we will be using a one instance variable for ordering

Using get_key

- Now we need to sort on the new key
- sorted(aList, key=get_key)
- aList.sort(key=get_key)
- get_key is the name of the function
  - get_key() calls the get key function
  - get_key without () is a ‘pointer’ to the get_key function
  - We want no parenthesis

So now lets sort

- Lets use that to sort a list of students.

Binary Search

- Now lets implement binary search
  - For the list of students

Basic Algorithms

- How would be get the sum of all the items in the list?
  - Without having the built in sum function to use?

Average

- How would we find the average GPA

Median Size

- Suppose we want the median size
  - Remember median from high school/middle school?
  - So what do we need to do to find the median?
  - So what do we need to do to find the median?
  - We need the list sorted – otherwise we can't find the middle
  - Lets look at that sorted list of students

Now we can find the median

- Now we can find the median size image
  - If even number lets just use the first of the two middle numbers
  - yes I just made the math faculty twitch in terror