Ruby Explorations VIII

Mark Keane...CSI...UCD

A Program

A: Six Degrees of Kevin Bacon

B: six degrees: up close and scrapie

C: the algorithm

D: program design

E: a nice trick with UriHacker module

F: the thing running

Part A:

Six degrees of separation...The Game

http://www.sixdegrees.org/

http://oracleofbacon.org/

Kevin Bacon

trivia game to find film that Richard Dreyfus and Kevin Bacon were in...

six degrees is a lot... 2/3 can easily connect several million people

play it using Ruby?

Kevin Bacon & Richard Dreyfuss

THE ORACLE OF BACON

Richard Dreyfuss has a Bacon number of 2.

Find a different link

Richard Dreyfuss	
Tin Men (1987)	
David DeBoy	
My One and Only (2009)	
Kevin Bacon	

How good a center is   Kevin Bacon   ?

Kevin Bacon Number	# of People
0	1
1	2333
2	236985
3	747329
4	184725
5	12551
6	1123
7	158
8	19

Kevin Bacon   to   Richard Dreyfuss   Find link   More options >>

Total number of linkable actors: 1185224
Weighted total of linkable actors: 3527941
Average Kevin Bacon number: 2.977

Six Degrees Design

less about Objects more about Web Scraping

to play the game, we need to know:

Kevin's films and the actors in them

the other films those actors were in...

the other actors in those films...and so on

how ?

Part B:

Six degrees of separation...Up close and scrapie

Six Degrees Basics

scrape wikipedia for all info...

two basic methods:

find_films_by_actor: that finds the films that an actors has taken part in

find_actors_in_film: that finds the actors that starred in a particular film

that's the essence of the Web interface

Six Degrees Basics

find_films_by_actor: goes to an actor's wikipedia page and finds the films that he/she is in

find_actors_in_film: goes to a film wikipedia page and finds the actors that star in that film

---

The Page

The Source

heading

table

film name

film link

useful txt

---

find_films_by_actor (imperfect)

assume you have parsed wikipage from Nokogiri
find the key bit listing the films (what is the key bit?)
extract the links from that bit (what is that?)
clean up what you are getting (like how?)

```ruby
def find_films_by_actor(doca, out = [])
  puts "Entering find_films_by_actor with #{find_name_title(doca)}."
  all_links = doca.search('//a[@href]')
  all_links.each do |link|
    link_info = link['href']
    if link_info.include?("(film)") &&
       !(link_info.include?("Category:") || link_info.include?("php"))
    then out << link_info end
  end
  out.uniq.collect {|link| strip_out_name(link)}
end
```

wikihac.rb

Some Problems

```
uri4 = "http://en.wikipedia.org/wiki/Saoirse_Ronan"
doc = Nokogiri::HTML(open(uri4, :allow_redirections
=> :safe))
```

wikihac.rb

markkean% ruby wikihac.rb

Checking the following: Saoirse_Ronan

Given non-wiki uri in make_wiki_uri: /w/index.php?
title=Hanna_(film)&action=edit&redlink=1

["http://en.wikipedia.org/wiki/Atonement_(film)",
"http://en.wikipedia.org/wiki/The_Lovely_Bones_(film)",
"http://en.wikipedia.org/wiki/Veronica_Guerin_(film)",
"/w/index.php?title=Hanna_(film)&action=edit&redlink=1"]

But...

Year	Film	Role	
2003-2004	The Clinic	Rhiannon Geraghty	4 Episodes
2005	Proof	Orla Boland	4 Episodes
	The Christmas Miracle of Jonathan Toomey	Celia Hardwick	
2007	Atonement	Briony Tallis (age 13)	Irish Film & Television A... Las Vegas Film Critics S... Phoenix Film Critics Soc... Female Nominated — BAFTA A... Nominated — Empire A... Nominated — Film Critic... Nominated — Golden G... Nominated — London F... Nominated — Online Fil... Nominated — Satellite A... Nominated — St. Louis A... Nominated — Young Arti... Nominated — Broadcast...
	I Could Never Be Your Woman	Izzie	
2008	Death Defying Acts	Benji	Irish Film & Television A... Nominated — Broadcast...
	City of Ember	Lina Mayfleet	Nominated — Irish Film...
2009	The Lovely Bones	Susie Salmon	Las Vegas Film Critics S... Phoenix Film Critics Soc... Female Broadcast Film Critics A... Irish Film & Television A... Nominated — BAFTA A... Nominated — Broadcast... Nominated — Detroit Fil... Nominated — Houston F... Nominated — Saturn Aw... Nominated — Young Arti...
	The True Confessions of Charlotte Doyle	Charlotte Doyle	
2010	The Way Back	Irena	post-production

Some Other Methods

```
def find_name_title(doca)
  if !doca.nil?
    then title = doca.search("//title").inner_text
    title.gsub(" - Wikipedia, the free encyclopedia", "")
  end
end

def strip_out_name(name, nameout = [])
    case
    when name =~ /^http:\/\/en.wikipedia.org\/wiki\//
        nameout = name.gsub(/http:\/\/en.wikipedia.org\/wiki\//, "")
    when name =~ /^en.wikipedia.org\/wiki\//
        nameout = name.gsub(/en.wikipedia.org\/wiki\//, "")
    when name =~ /^\/wiki\//
        nameout = name.gsub(/\/wiki\//, "")
    when name !=~ URI::regexp || name.nil?
      print "URI is not right in STRIP_OUT_NAME: "; p "#{@in_uri}"
    end
  nameout
end
```

wikihac.rb

Six Degrees Basics

find_films_by_actor: goes to an actor's wikipedia page and finds the films that he/she is in

find_actors_in_film: goes to a film wikipedia page and finds the actors that star in that film

---

The Page

The Page

The Source

table
useful txt
actor link
heading
actor name

---

find_actors_in_film (imperfect)

```ruby
def find_actors_in_film(doc, out = [])
  puts "     ***** Entering find_actors_in_film with #{@title}."
  all_tables = doc.search("//tr")
  all_tables.each do |tr_node|
    tr_childs_less_newline_nodes =
        tr_node.children.select{|node| !(node.inner_text == "\n")}
    starring_node = tr_childs_less_newline_nodes[0]
    if starring_node.inner_text == "Starring"
      then rest_of_childs = tr_childs_less_newline_nodes[1].children
        rest_of_childs.each {|node| if node['href'] then out << node['href'] end}
      end
    end
  end
  out.uniq.collect {|link| link.strip_out_name}
end
```

wikihac.rb

---

find_actors_in_film: algorithm

```ruby
uri8 = "http://en.wikipedia.org/wiki/Veronica_Guerin_(film)"
doc = Nokogiri::HTML(open(uri8,:allow_redirections => :safe))
find_actors_in_film(doc)
```

pokewiki.min.rb

$ruby wikihac.rb

Checking the following: Veronica_Guerin_(film)
["http://en.wikipedia.org/wiki/Cate_Blanchett",   aka Spodo Komodo
"http://en.wikipedia.org/wiki/Gerard_McSorley",
"http://en.wikipedia.org/wiki/Ciar%C3%A1n_Hinds",
"http://en.wikipedia.org/wiki/Brenda_Fricker"]
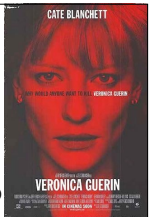
Part C:
Six Degrees of Separation...an Algorithm

Six Degrees: Algorithm

L0: find all the films that KB is in

   is RD a co-star in any of those films ?   Kevin Bacon (KB)

     If yes, stop and say degree is 0   Richard Dreyfuss (RD)

     if no, proceed to next level

L1: find all films of KB's co-stars are in (call them CO-1)

   is RD a co-star in any of their (ie CO-1) films

     if yes then, stop and say degree is 1

     if no then proceed to next level

L2: find all films of co-stars of CO-1 (call them CO-2)

   is RD a co-star in any of their (ie CO-2) films

     if yes, then stop and say degree is 2

     if no, then proceed to next level....

L0 test: is RD a co-star of KB this film?
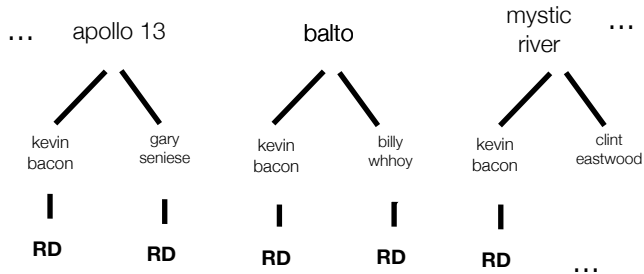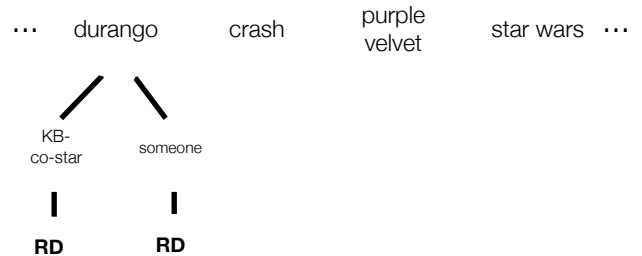
... apollo 13   balto   mystic river ...

kevin bacon   gary seniese   kevin bacon   billy whhoy   kevin bacon   clint eastwood

RD   **RD**   **RD**   **RD**   **RD**   ...

Ln test: is RD a co-star of KB-co-star from previous level ?

... durango   crash   purple velvet   star wars ...

KB-co-star   someone

RD   **RD**

Ln test: is RD a co-star of KB-co-star from previous level ?

... durango   crash   purple velvet   star wars ...

KB-co-star   someone

RD   **RD**

Ln test: is RD a co-star of KB-co-star from previous level ?

... durango   crash   purple velvet   star wars ...

KB-co-star   someone

RD   **RD**
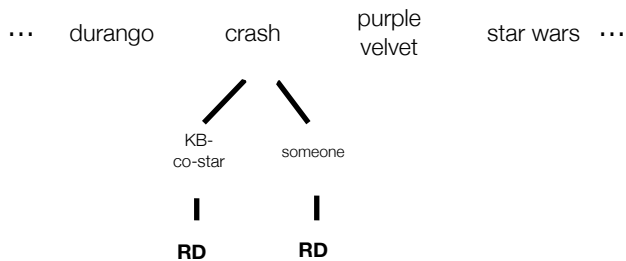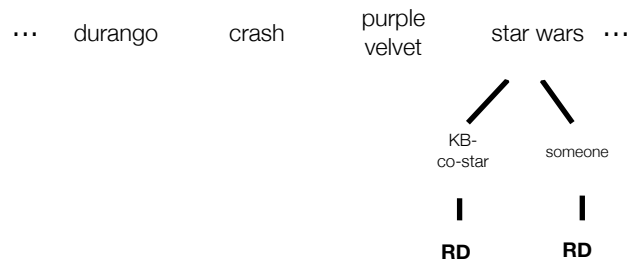
Working within any level...e.g., L1

kevin bacon   richard dreyfuss

the films of kevin_bacon

L1   ...   [ ] L1

L1   L1

L1   L1

L1   L1

L1   L1

L1   [ ]   ...   L1

films   actors   the co-actors of kevin_bacon for the next level

---
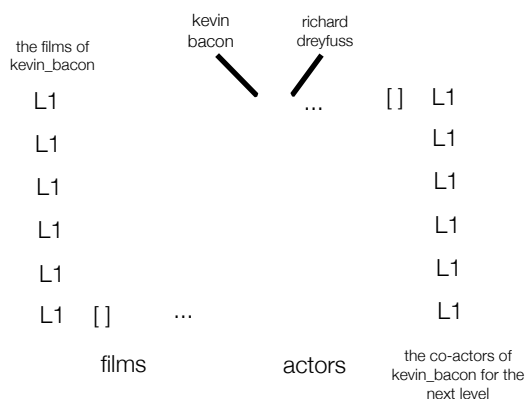
Program will use breadth-first search of films...

films

L1

L2

L3

L4

L5 ...
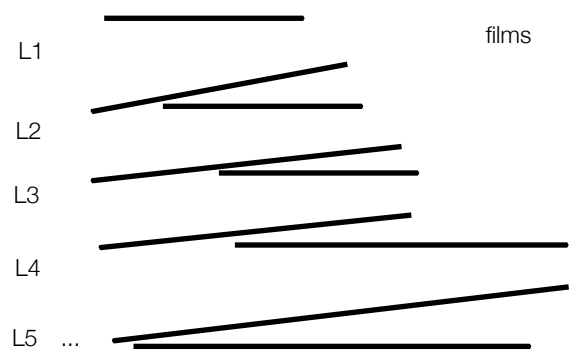
Part D:

Six degrees of separation...the program design

Six Degrees Implement

Break the wikihac.rb file up into loads of different classes

Sixtop.rb   => top level of program;
                       load_web_page method
                       call to start SixDegrees

Wikisearcher.rb => class that finds films and actors

WikiLoad.rb   => class handles download errors

UriHacker.rb   => module for hacking uris

SixDegrees.rb   => class that searches tru pages

Six Degrees Implement

2 classes and 2 modules: **SixDegrees** for game instances and **Wikisearcher, WikiLoader, UriHacker** for scraping wikipedia

SixDegrees, will hold a set of key data structures that will handle the arrays of films and actors, level count, what we are searching for and so on...

Wikisearcher will do **find_films_by_actor** and **find_actors_in_film,** along with other bits for hacking web addresses

NB; at the interface we print names "Kevin_Bacon", but internally we will be passing around links as strings: "http://en.wikipedia.org/wiki/Kevin_Bacon"

Wikiloader Class: A fat predicate

checking a web page is loadable...

is not straight-forward...there are lots of errors

some are HTTP errors, some are socket errors, some arise from giving a bad link or timeouts

some seem to arise from the speed of access

and...you also can get locked out of Wikipedia site if you bombard it with too many requests, too quickly

so, I will only give you half the solution, in case you break the internet !

---

sixtop

wikiloader

urihacker
module

wikisearch
module

sixdegrees

WikiLoader v.1

```
def load_uri(uri, flag)
    puts "\nChecking the following: #{strip_out_name(uri)}"
    begin
  doc = open(uri, :allow_redirections => :safe)
  rescue OpenURI::HTTPError => the_error
   puts "Error, found bad status code: #{the_error}"
    end
    if the_error.instance_of?(OpenURI::HTTPError)
        then puts "...so, we are not doing #{uri}"
    else doc = Nokogiri::HTML(open(uri, :allow_redirections => :safe))
        if flag == "actor" then out = find_films_by_actor(doc)
        elsif flag == "film" then out = find_actors_in_film(doc)
        end
    end
  out
end
```

wikihac.rb

WikiLoader v.1

```
def load_uri(uri, flag)
    puts "\nChecking the following: #{strip_out_name(uri)}"
  begin
          begin
  doc = open(uri, :allow_redirections => :safe)     error-type 1
rescue OpenURI::HTTPError => the_error
  puts "Error, found bad status code: #{the_error}"
  end           end
    if the_error.instance_of?(OpenURI::HTTPError)
        then puts "...so, we are not doing #{uri}"
    else doc = Nokogiri::HTML(open(uri, :allow_redirections => :safe))
        if flag == "actor" then out = find_films_by_actor(doc)
        elsif flag == "film" then out = find_actors_in_film(doc)
        end
    end
  out
end
```

wikihac.rb

WikiLoader v.1

```
def load_uri(uri, flag)
    puts "\nChecking the following: #{strip_out_name(uri)}"
  begin
  doc = open(uri, :allow_redirections => :safe)
 rescue OpenURI::HTTPError => the_error
  puts "Error, found bad status code: #{the_error}"
  end
    if the_error.instance_of?(OpenURI::HTTPError)     if everything is
        then puts "...so, we are not doing #{uri}"        ok
    else doc = Nokogiri::HTML(open(uri, :allow_redirections => :safe))
        if flag == "actor" then out = find_films_by_actor(doc)
        elsif flag == "film" then out = find_actors_in_film(doc)
        end
    end
  out
end
```

wikihac.rb

WikiLoader v.2

```
module WikiLoader
  def self.get_page?(uri, out = [])
    puts "\nChecking the following: #{uri}"
    begin
        doc = open(uri, :allow_redirections => :safe)
    rescue OpenURI::HTTPError => the_error
      puts "Errror, bad status code: #{the_error}"
    rescue Errno::ENOENT => the_error
          puts "Error ENOCENT code: #{the_error}"
    rescue Errno::EHOSTUNREACH => the_error
          puts "Error, request for a page #{uri} was unreachable."
    rescue SocketError => the_error
          puts "Error, request for a page #{uri} threw a SocketError."
    rescue Errno::ETIMEDOUT => the_error
        puts "Error, request for a page #{uri} threw a TimeoutError."
    end
    if the_error
   then puts "...so, we are not doing #{uri}";
        return false
    else true end
  end
end
```

wikiloader.rb

Six Degrees Design

we have a single object that has slots for all the things we will handle, actors, films, degrees etc...

we populate with with KB and RD and set it off

it recursively calls a single method and loops down until it finds a match

longest run, around 2-3 hours...several thousand films and actors checked

...but, it still hits I/O errors, even ones that should be handled ?

The SixDegrees Class I

```
class SixDegrees
  attr_accessor :origin, :target, :base_actor,
                :film_stack,:curr_film,:curr_actors,
                :this_level_actors,:next_level_actors,:degree

  def initialize(base, target)
    @origin = base
    @target = target.add_full_uri
    @base_actor = base.add_full_uri
    @film_stack  = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @this_level_actors  = [@base_actor]
    @next_level_actors  = []
    @degree = 0
  end
...
```

sixdegrees.rb

```
sixo = SixDegrees.new("Kevin_Bacon", "Richard_Dreyfus")
sixo.check_level
```

sixto n.rb

```
class SixDegrees

attr_accessor ...
def check_level
  if @curr_actors.include?(@target) || @base_actor == @target
    then puts "\n\n"
    puts "###################################################"
    puts "#{@base_actor.strip_out_name} and #{@target.strip_out_name}"
    puts " are at  #{@degree} degrees from #{@origin}."
    puts "They were both in #{@curr_film.strip_out_name}."
    puts "###################################################"
    puts "\n\n"
  elsif @this_level_actors.empty? && @film_stack.empty?
    then p "Using test1"
    @this_level_actors = (@next_level_actors.uniq).select
                                    {|actor| !actor.eql?(@base_actor)}
    @next_level_actors = []
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors)
    @degree += 1
    check_level
  ...
                                                    sixdegrees.rb
```

```
class SixDegrees

attr_accessor ...                      cond1: match found
def check_level
  if @curr_actors.include?(@target) || @base_actor == @target
    then puts "\n\n"
    puts "###################################################"
    puts "#{@base_actor.strip_out_name} and #{@target.strip_out_name}"
    puts " are at  #{@degree} degrees from #{@origin}."
    puts "They were both in #{@curr_film.strip_out_name}."
    puts "###################################################"
    puts "\n\n"
  elsif @this_level_actors.empty? && @film_stack.empty?action1: print result
    then p "Using test1"
    @this_level_actors = (@next_level_actors.uniq).select
                                    {|actor| !actor.eql?(@base_actor)}
    @next_level_actors = []
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors)
    @degree += 1
    check_level
  ...
                                                    sixdegrees.rb
```

```
class SixDegrees

attr_accessor ...
def check_level
  if @curr_actors.include?(@target) || @base_actor == @target
    then puts "\n\n"
    puts "###################################################"
    puts "#{@base_actor.strip_out_name} and #{@target.strip_out_name}"
    puts " are at  #{@degree} degrees from #{@origin}."
    puts "They were both in #{@curr_film.strip_out_name}."
    puts "###################################################"
    puts "\n\n"           cond2: both lists empty
  elsif @this_level_actors.empty? && @film_stack.empty?
    then p "Using test1"
    @this_level_actors = (@next_level_actors.uniq).select
                                    {|actor| !actor.eql?(@base_actor)}
    @next_level_actors = []
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors)
    @degree += 1
    check_level                action2: move to next level
  ...
                                                    sixdegrees.rb
```

```
class SixDegrees

attr_accessor ...
def check_level
  ...
  elsif (@this_level_actors.first == @base_actor)
        && @film_stack.empty?
        && @this_level_actors.size == 1
then p "Using test2"
    @this_level_actors = (@next_level_actors.uniq).select
                                    {|actor| !actor.eql?(@base_actor)}
    @next_level_actors = []
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors)
    @degree += 1
    check_level
  elsif !@this_level_actors.empty? && @film_stack.empty?
    then p "Using test3"
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors.select
      {|actor| !actor.eql?(@base_actor)}
  check_level                                       sixdegrees.rb
```

```
class SixDegrees

attr_accessor ...
def check_level
  ...                      cond3: only the base_actor is left
  elsif (@this_level_actors.first == @base_actor)     and film list is empty
        && @film_stack.empty?
        && @this_level_actors.size == 1
then p "Using test2"
    @this_level_actors = (@next_level_actors.uniq).select
                                    {|actor| !actor.eql?(@base_actor)}
    @next_level_actors = []
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors)
    @degree += 1
    check_level                     action3: move to next level
  elsif !@this_level_actors.empty? && @film_stack.empty?
    then p "Using test3"
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors.select
      {|actor| !actor.eql?(@base_actor)}
  check_level                                       sixdegrees.rb
```

```
class SixDegrees
attr_accessor ...
def check_level
  ...
  elsif (@this_level_actors.first == @base_actor)
        && @film_stack.empty?
        && @this_level_actors.size == 1
then p "Using test2"
    @this_level_actors = (@next_level_actors.uniq).select
                                    {|actor| !actor.eql?(@base_actor)}
    @next_level_actors = []
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors)
    @degree += 1            cond4: some actors but no films
    check_level
  elsif !@this_level_actors.empty? && @film_stack.empty?
    then p "Using test3"
    @base_actor = @this_level_actors.pop
    @film_stack = load_web_page(@base_actor, :actor)
    @curr_film = @film_stack.pop
    @curr_actors = load_web_page(@curr_film, :film)
    @next_level_actors.concat(@curr_actors.select
      {|actor| !actor.eql?(@base_actor)}
  check_level                                       sixdegrees.rb
            action4: move along this level
```

```
class SixDegrees

 attr_accessor ...
def check_level
  ...
  elsif @this_level_actors.empty? && !@film_stack.empty?
 then p "Using test4"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
   check_level
 elsif !@this_level_actors.empty? && !@film_stack.empty?
    then p "Using test5"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
     check_level
 else puts "woops...fall through in check_level"
 end
end
```
sixdegrees.rb

```
class SixDegrees

 attr_accessor ...
def check_level                    cond5: no actors, but some films
  ...
  elsif @this_level_actors.empty? && !@film_stack.empty?
 then p "Using test4"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
   check_level                     action5: get some actors
 elsif !@this_level_actors.empty? && !@film_stack.empty?
    then p "Using test5"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
     check_level
 else puts "woops...fall through in check_level"
 end
end
```
sixdegrees.rb

```
class SixDegrees

 attr_accessor ...
def check_level
  ...
  elsif @this_level_actors.empty? && !@film_stack.empty?
 then p "Using test4"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
   check_level             cond6: some actors, some films
 elsif !@this_level_actors.empty? && !@film_stack.empty?
    then p "Using test5"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
     check_level
 else puts "woops...fall through in check_level"   action6: pop a film and get its actors
 end
end
```
test5 is used the most   *sixdegrees.rb*

```
class SixDegrees

 attr_accessor ...
def check_level
  ...
  elsif @this_level_actors.empty? && !@film_stack.empty?
 then p "Using test4"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
   check_level
 elsif !@this_level_actors.empty? && !@film_stack.empty?
    then p "Using test5"
     @curr_film = @film_stack.pop
     @curr_actors = load_web_page(@curr_film, :film)
     @next_level_actors.concat(@curr_actors.select
            {|actor| !actor.eql?(@base_actor)})
     check_level
 else puts "woops...fall through in check_level"
 end
end                   action7: the woops option
```
woops is never used...   *sixdegrees.rb*

Part E:

I love this...see what I did with UriHacker module

UriHacker Module

```
module Urihacker
  def strip_out_name
   if self =~ /^http:\/\/en.wikipedia.org\/wiki\//
    then name = self.gsub(/http:\/\/en.wikipedia.org\/wiki\//, "")
   elsif self =~ /^en.wikipedia.org\/wiki\//
    then name = self.gsub(/en.wikipedia.org\/wiki\//, "")
   elsif self =~ /^\/wiki\//
    then name = self.gsub(/\/wiki\//, "")
   elsif !(self =~ URI::regexp)
    then print "URI looks bad; can't strip name: "; p "#{self}"; name = self end
   name
   end

  def add_tag(tag)
   if tag == :actor
    then return self + "_(actor)"
   elsif tag == :film
    then return self + "_(film)" end
  end
  def add_full_uri
     "http://en.wikipedia.org/wiki/" + self
  end
...
 end
```
urihacker.rb

Stick it here...

```
class String
  include Urihacker
end
```

*this has huge benefits
as any uri string can
be hacked by the methods
of this module; but watch for
strip_out_name's
problems with **nil** and **[ ]***

sixtop.rb

Other things in sixtop.rb

```ruby
def load_web_page(uri, flag, out = [])
  load_ok_flag = if uri.nil? then false else WikiLoader.get_page?(uri) end
  if !load_ok_flag
   then "just pass on..."
  elsif load_ok_flag && flag == :actor
   then doc_parse = Nokogiri::HTML(open(uri))
      search_obj = WikiSearcher.new(doc_parse)
      new_actors = search_obj.find_films_by_actor
      p new_actors.uniq.compact
      out.concat(new_actors)
  elsif load_ok_flag && flag == :film
   then doc_parse = Nokogiri::HTML(open(uri))
      search_obj = WikiSearcher.new(doc_parse)
      new_films = search_obj.find_actors_in_film
      p new_films.uniq.compact
      out.concat(new_films)
  end
  out.uniq.compact.collect {|name| name.add_full_uri}
end
```

sixtop.rb

---

sixtop

wikiloader

urihacker
module

wikisearch
module

sixdegrees

Part F:

Six degrees of separation...running

```ruby
sixo = SixDegrees.new("Kevin_Bacon", "Richard_Dreyfus")
sixo.check_level
```

sixtop.rb

```
Checking the following: http://en.wikipedia.org/wiki/Kevin_Bacon
Entering find_films_by_actor with Kevin Bacon.
["Diner_(film)", "Footloose_(1984_film)", "A_Few_Good_Men_(film)", "JFK_(film)", "Apollo_13_(film)",
"Mystic_River_(film)", "Trapped_(2002_film)", "Friday_the_13th_(1980_film)", "Tremors_(film)", "Frost/
Nixon_(film)", "The_Big_Picture_(film)", "Sleepers_(film)", "Picture_Perfect_(1997_film)",
"Telling_Lies_in_America_(film)", "Loverboy_(2005_film)", "Starting_Over_(1979_film)", "The_Gift_(1979_film)",
"Only_When_I_Laugh_(film)", "Quicksilver_(film)", "End_of_the_Line_(1987_film)", "Criminal_Law_(film)",
"Balto_(film)", "My_Dog_Skip_(film)", "Novocaine_(film)", "Death_Sentence_(2007_film)",
"My_One_and_Only_(film)", "Elephant_White_(film)", "Super_(2011_film)", "Mister_Roberts_(1984_film)"]

Checking the following: http://en.wikipedia.org/wiki/Mister Roberts (1984 film)
Entering find_actors_in_film with Mister Roberts (1984 film).
["Robert_Hays", "Kevin_Bacon", "Raye_Birk", "Charles_Durning", "Marilu_Henner", "Howard_Hesseman",
"Peter_Jason", "Charles_Lang", "Christopher_Murray", "Joe_Pantoliano"]
At level 0, about to consider Mister_Roberts_(1984_film) with Kevin_Bacon ...
There are 1 actors remaining at level 0 to be checked.
There are 0 actors at next level 1.
There are 28 films at this level 0 for the actor we are checking.
["http://en.wikipedia.org/wiki/Robert_Hays", "http://en.wikipedia.org/wiki/Kevin_Bacon", "http://
en.wikipedia.org/wiki/Raye_Birk", "http://en.wikipedia.org/wiki/Charles_Durning", "http://en.wikipedia.org/
wiki/Marilu_Henner", "http://en.wikipedia.org/wiki/Howard_Hesseman", "http://en.wikipedia.org/wiki/
Peter_Jason", "http://en.wikipedia.org/wiki/Charles_Lang", "http://en.wikipedia.org/wiki/Christopher_Murray",
"http://en.wikipedia.org/wiki/Joe_Pantoliano"]
"http://en.wikipedia.org/wiki/Kevin_Bacon"
"http://en.wikipedia.org/wiki/Richard_Dreyfuss"
"Using test5"

Checking the following: http://en.wikipedia.org/wiki/Super (2011 film)
Entering find_actors_in_film with Super (2010 American film).
["Rainn_Wilson", "Ellen_Page", "Liv_Tyler", "Kevin_Bacon", "Nathan_Fillion", "Michael_Rooker"]
At level 0, about to consider Super_(2011_film) with Kevin_Bacon ...
There are 1 actors remaining at level 0 to be checked.
There are 5 actors at next level 1.
There are 27 films at this level 0 for the actor we are checking.
["http://en.wikipedia.org/wiki/Rainn_Wilson", "http://en.wikipedia.org/wiki/Ellen_Page", "http://
en.wikipedia.org/wiki/Liv_Tyler", "http://en.wikipedia.org/wiki/Kevin_Bacon", "http://en.wikipedia.org/wiki/
Nathan_Fillion", "http://en.wikipedia.org/wiki/Michael_Rooker"]
```

```
Checking the following: http://en.wikipedia.org/wiki/Point_of_Origin_(film)
Entering find_actors_in_film with Point of Origin (film).
[]
At level 1, about to consider Point_of_Origin_(film) with Illeana_Douglas ...
There are 59 actors remaining at level 1 to be checked.
There are 3838 actors at next level 2.
There are 17 films at this level 1 for the actor we are checking.
[]
"http://en.wikipedia.org/wiki/Illeana Douglas"
"http://en.wikipedia.org/wiki/Richard Dreyfuss"
"Using test5"

Checking the following: http://en.wikipedia.org/wiki/Lansky_(film)
Entering find_actors_in_film with Lansky (film).
["Richard_Dreyfuss", "Eric_Roberts", "Anthony_LaPaglia"]
At level 1, about to consider Lansky_(film) with Illeana_Douglas ...
There are 59 actors remaining at level 1 to be checked.
There are 3845 actors at next level 2.
There are 15 films at this level 1 for the actor we are checking.
["http://en.wikipedia.org/wiki/Richard Dreyfuss", "http://en.wikipedia.org/wiki/Eric Roberts", "http://
en.wikipedia.org/wiki/Anthony LaPaglia"]
"http://en.wikipedia.org/wiki/Illeana Douglas"
"http://en.wikipedia.org/wiki/Richard Dreyfuss"

###########################################################
Illeana_Douglas and Richard_Dreyfuss are at  1 degrees from Kevin_Bacon.
They were both in Lansky_(film).
###########################################################
```

Conclusions

formulate questions for next-week look-back lecture;
submit them in advance by email
mark.keane@ucd.ie (or on Moodle)

when we move to Rails, will assume to know all of
this

**Mark. If you have time go to next week's
lectures of previous issues raised...**