

# 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](#)



[Kevin Bacon](#)

to [Richard Dreyfuss](#)

[Find link](#)

[More options >>](#)

How good a center is  ?

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

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

# Music

In 1995, Bacon formed a band called [The Bacon Brothers](#).

## Filmography

### Films

Year	Film	Role
1978	<a href="#">National Lampoon's Animal House</a>	Chip Diller
1979	<a href="#">Starting Over</a>	Husband
1979	<a href="#">The Gift</a>	Teddy
1980	<a href="#">Hero at Large</a>	2nd Teenager
1980	<a href="#">Friday the 13th</a>	Jack Burrell
1981	<a href="#">Only When I Laugh</a>	Don
1982	<a href="#">Diner</a>	Timothy Fenwick
1982	<a href="#">Forty Deuce</a>	Ricky
1983	<a href="#">Enormous Changes at the Last Minute</a>	Dennis
1984	<a href="#">Footloose</a>	Ren McCormack
1986	<a href="#">Quicksilver</a>	Jack Casey
1987	<a href="#">White Water Summer</a>	Vic
1987	<a href="#">End of the Line</a>	Everett
1987	<a href="#">Planes, Trains &amp; Automobiles</a>	Taxi Racer
1988	<a href="#">She's Having a Baby</a>	Jefferson 'Jake' Edward Brions



WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article

Interaction  
About Wikipedia  
Community portal  
Recent changes  
Contact Wikipedia  
Donate to Wikipedia  
Help

Toolbox  
Print/export  
Languages  
العربية  
Български  
Català  
Dansk  
Deutsch  
Español  
Euskara  
فارسی  
Français  
한국어  
Hrvatski  
Ido  
Bahasa Indonesia  
Italiano  
עברית  
Magyar  
Nederlands  
日本語

Article Discussion

Read View source View history

Search

## Kevin Bacon

From Wikipedia, the free encyclopedia

For other uses, see [Kevin Bacon \(disambiguation\)](#).



This **biographical article** needs additional citations for verification. Please help by adding reliable sources. Contentious material about living persons that is unsourced or poorly sourced must be removed immediately, especially if potentially libelous or harmful. (January 2010)

**Kevin Norwood Bacon** <sup>[1]</sup> (born July 8, 1958) is an American film and theater actor whose notable roles include *Animal House*, *Footloose*, *Flatliners*, *A Few Good Men*, *Apollo 13*, *Mystic River*, *The Woodsman*, *Friday the 13th*, *Hollow Man*, and *Tremors*.

Bacon has won Golden Globe and Screen Actors Guild Awards, was nominated for an Emmy Award, <sup>[2]</sup> and was named by *The Guardian* as one of the best actors never to have received an Academy Award nomination. <sup>[3]</sup>

In 2003, Bacon received a star on the Hollywood Walk of Fame. <sup>[2]</sup>

Contents [hide]  
[1 Early life](#)  
[2 Acting career](#)  
[3 Personal life](#)  
[4 Six Degrees of Kevin Bacon](#)  
[5 Music](#)  
[6 Filmography](#)  
[6.1 Films](#)  
[6.2 Television](#)  
[7 References](#)  
[8 External links](#)

### Early life

Bacon, one of six children, was born and raised in a close-knit family in Philadelphia, Pennsylvania. His mother, Ruth Hilda (née Holmes; 1916–1991), taught elementary school and was a liberal activist, while his father, Edmund Bacon, was a well-respected architect. At 16, Bacon attended the Pennsylvania Governors School for the Arts, a state-funded five-week arts program which helped solidify Bacon's passion for the arts. [citation needed]

### Acting career



Kevin Bacon at the Screen Actors Guild Awards, Shrine Auditorium, Los Angeles, on January 23rd, 2010.

**Born** Kevin Norwood Bacon  
July 8, 1958 (age 51)  
Philadelphia, Pennsylvania  
**Occupation** Actor  
**Years active** 1978–present  
**Spouse(s)** Kyra Sedgwick (1988–present)  
[Official website](#)

The Page

# The Source



```
wgCurRevisionId=362508044,  
wgVersion="1.16wmf4",  
wgEnableAPI=true,  
wgEnableWriteAPI=true,  
wgSeparatorTransformTable=[ "", "" ],  
wgDigitTransformTable=[ "", "" ],  
wgMainPageTitle="Main Page",  
wgFormattedNamespaces={ "-2": "Media", "-1": "Special", "0": "", "1": "Talk",  
wgNamespaceIds={ "media": -2, "special": -1, "" : 0, "talk": 1, "user": 2, "talkback": 3 }
```

# 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?)

```
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
```

# 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...



## Filmography

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

# Some Other Methods

```
def find_name_title(doc)
  if !doc.nil?
    then title = doc.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



# Close Encounters of the Third Kind

From Wikipedia, the free encyclopedia

***Close Encounters of the Third Kind***, also known as *CE3K*, is a 1977 American science fiction film written and directed by Steven Spielberg. The film stars Richard Dreyfuss, François Truffaut, Melinda Dillon, Teri Garr, Bob Balaban, and Cary Guffey. It tells the story of Roy Neary, an Indiana electrical lineman, whose life changes after he has an encounter with an unidentified flying object. In addition, the United States government is also aware of the UFOs as is a team of international scientific researchers.

*Close Encounters* was a long-cherished project for Spielberg. In late 1973, he developed a deal with Columbia Pictures for a science fiction film. Though Spielberg receives sole credit for the script, he was assisted by Paul Schrader, John Hill, David Giler, Hal Barwood, Matthew Robbins, and Jerry Belson, all of whom contributed to the screenplay in varying degrees. The title is derived from astronomer/ufologist J. Allen Hynek's classification of close encounters with aliens, in which the third kind denotes human observations of actual aliens or "animate beings".

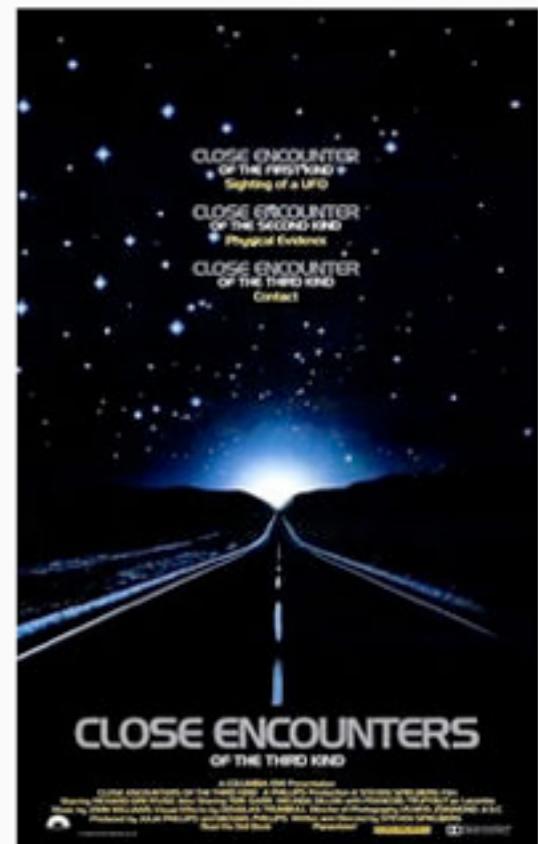
Filming began in May 1976. Douglas Trumbull served as the visual effects supervisor, while Carlo Rambaldi designed the aliens. *Close Encounters* was released in November 1977 and was a critical and financial success. The film was reissued in 1980 as *Close Encounters of the Third Kind: The Special Edition*, which featured additional scenes. A third cut of the film was released to home video (and later DVD) in 1998. The film received numerous awards and nominations at the 50th Academy Awards, 32nd British Academy Film Awards, the Saturn Awards and has been widely acclaimed by the American Film Institute. In December 2007, it was deemed "culturally, historically, or aesthetically significant" by the United States Library of Congress and selected for preservation in the National Film Registry.<sup>[1]</sup>

## Contents [hide]

- 1 Plot
- 2 Cast
- 3 Production
  - 3.1 Development
  - 3.2 Filming
  - 3.3 Visual effects
  - 3.4 Post-production
- 4 Soundtrack
  - 4.1 Musical score
- 5 Themes and antecedents
- 6 Reaction
  - 6.1 Release

The Page

## Close Encounters of the Third Kind



Directed by	Steven Spielberg
Produced by	Julia Phillips Michael Phillips
Written by	Steven Spielberg
Starring	Richard Dreyfuss François Truffaut Melinda Dillon Terri Garr
Music by	John Williams
Cinematography	Vilmos Zsigmond
Editing by	Michael Kahn
Distributed by	Columbia Pictures
Release date(s)	November 16, 1977



# Close Encounters of the Third Kind

From Wikipedia, the free encyclopedia

***Close Encounters of the Third Kind***, also known as *CE3K*, is a 1977 American science fiction film written and directed by Steven Spielberg. The film stars Richard Dreyfuss, François Truffaut, Melinda Dillon, Teri Garr, Bob Balaban, and Cary Guffey. It tells the story of Roy Neary, an Indiana electrical lineman, whose life changes after he has an encounter with an unidentified flying object. In addition, the United States government is also aware of the UFOs as is a team of international scientific researchers.

*Close Encounters* was a long-cherished project for Spielberg. In late 1973, he developed a deal with Columbia Pictures for a science fiction film. Though Spielberg receives sole credit for the script, he was assisted by Paul Schrader, John Hill, David Giler, Hal Barwood, Matthew Robbins, and Jerry Belson, all of whom contributed to the screenplay in varying degrees. The title is derived from astronomer/ufologist J. Allen Hynek's classification of close encounters with aliens, in which the third kind denotes human observations of actual aliens or "animate beings".

Filming began in May 1976. Douglas Trumbull served as the visual effects supervisor, while Carlo Rambaldi designed the aliens. *Close Encounters* was released in November 1977 and was a critical and financial success. The film was reissued in 1980 as *Close Encounters of the Third Kind: The Special Edition*, which featured additional scenes. A third cut of the film was released to home video (and later DVD) in 1998. The film received numerous awards and nominations at the 50th Academy Awards, 32nd British Academy Film Awards, the Saturn Awards and has been widely acclaimed by the American Film Institute. In December 2007, it was deemed "culturally, historically, or aesthetically significant" by the United States Library of Congress and selected for preservation in the National Film Registry.<sup>[1]</sup>

Contents [hide]	
1	Plot
2	Cast
3	Production
3.1	Development
3.2	Filming
3.3	Visual effects
3.4	Post-production
4	Soundtrack
4.1	Musical score
5	Themes and antecedents
6	Reaction
6.1	Release

The Page

## Close Encounters of the Third Kind



Directed by	Steven Spielberg
Produced by	Julia Phillips Michael Phillips
Written by	Steven Spielberg
Starring	Richard Dreyfuss François Truffaut Melinda Dillon Terri Garr
Music by	John Williams
Cinematography	Vilmos Zsigmond
Editing by	Michael Kahn
Distributed by	Columbia Pictures
Release date(s)	November 16, 1977

# The Source

```
<!-- /jumpto -->
<!-- bodytext -->
<table class="infobox vevent" cellspacing="5" style="width:22em; text-align:left; border:none; margin-left: auto; margin-right: auto;">
<tr>
<th colspan="2" class="summary" style="text-align:center; text-align:center; font-size:125%; font-weight:bold;">Cast
</tr>
<tr class="">
<td colspan="2" class="" style="text-align:center;">
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Directed by
<td class="description" style=""><a href="/wiki/Steven Spielberg" title="Steven Spielberg">Steven Spielberg</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Produced by
<td class="" style=""><a href="/wiki/Julia Phillips" title="Julia Phillips">Julia Phillips</a><br />
<a href="/wiki/Michael Phillips (producer)" title="Michael Phillips (producer)">Michael Phillips</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Written by
<td class="" style="">Steven Spielberg</td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Starring
<td class="" style=""><a href="/wiki/Richard Dreyfuss" title="Richard Dreyfuss">Richard Dreyfuss</a><br />
<a href="/wiki/François Truffaut" title="François Truffaut">François Truffaut</a><br />
<a href="/wiki/Melinda Dillon" title="Melinda Dillon">Melinda Dillon</a><br />
<a href="/wiki/Terri Garr" title="Terri Garr" class="mw-redirect">Terri Garr</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Music by
<td class="" style=""><a href="/wiki/John Williams" title="John Williams">John Williams</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Cinematography
<td class="" style=""><a href="/wiki/Vilmos Zsigmond" title="Vilmos Zsigmond">Vilmos Zsigmond</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Editing by
<td class="" style=""><a href="/wiki/Michael Kahn (film editor)" title="Michael Kahn (film editor)">Michael Kahn</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Distributed by
<td class="" style=""><a href="/wiki/Columbia Pictures" title="Columbia Pictures">Columbia Pictures</a></td>
</tr>
<tr class="">
<th style="text-align:left; white-space:nowrap; text-align:left;">Release date(s)
<td class="" style=""><span style="white-space: nowrap;">1977-06-10</td>
</tr>
```



# The Source



table

```
<!-- jumpTo -->
          <!-- bodytext -->
<table class="infobox vevent" cellspacing="5" style="width:22em; text-align:left; border-collapse: collapse; margin-left: auto; margin-right: auto;">
| Close Encounters of the Third Kind | |
| --- | --- |
|  | |
| Directed by | Steven Spielberg |
| Produced by | Julia Phillips Michael Phillips (producer) |
| Written by | Steven Spielberg |
| Starring | Richard Dreyfuss François Truffaut Melinda Dillon Terri Garr |
| Music by | John Williams |
| Cinematography by | Vilmos Zsigmond |
| Editing by | Michael Kahn (film editor) |
| Distributed by | Columbia Pictures |
| Release date(s) | (1977) |

```

actor link

useful txt

heading

actor name

# find\_actors\_in\_film (imperfect)

```
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

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

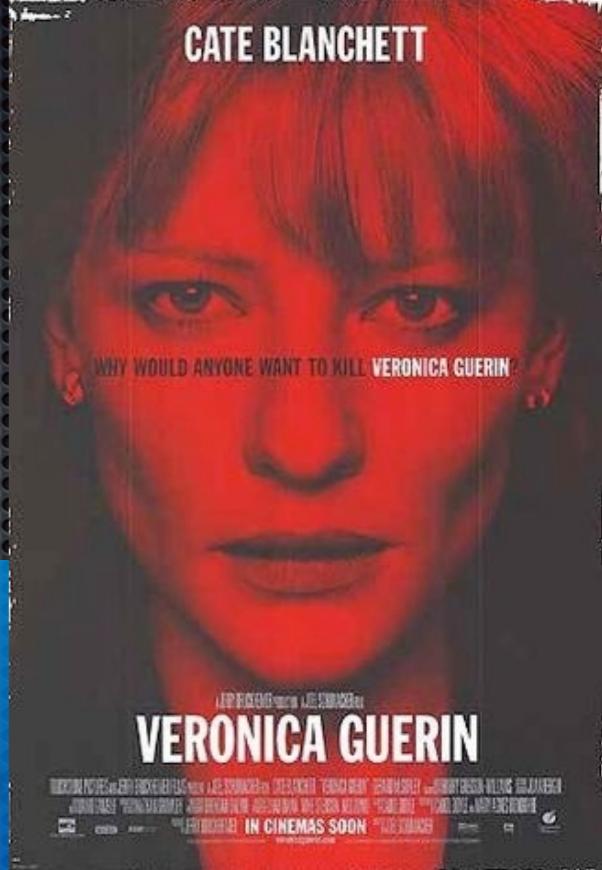
*pokewiki.min.rb*

\$ruby wikihac.rb

Checking the following: Veronica\_Guerin\_(film)

[[http://en.wikipedia.org/wiki/Cate Blanchett](http://en.wikipedia.org/wiki/Cate_Blanchett),  
[http://en.wikipedia.org/wiki/Gerard McSorley](http://en.wikipedia.org/wiki/Gerard_McSorley),  
[http://en.wikipedia.org/wiki/Ciar%C3%A1n Hinds](http://en.wikipedia.org/wiki/Ciar%C3%A1n_Hinds),  
[http://en.wikipedia.org/wiki/Brenda Fricker](http://en.wikipedia.org/wiki/Brenda_Fricker)]

aka Spodo Komodo



# Part C:

## Six Degrees of Separation...an Algorithm

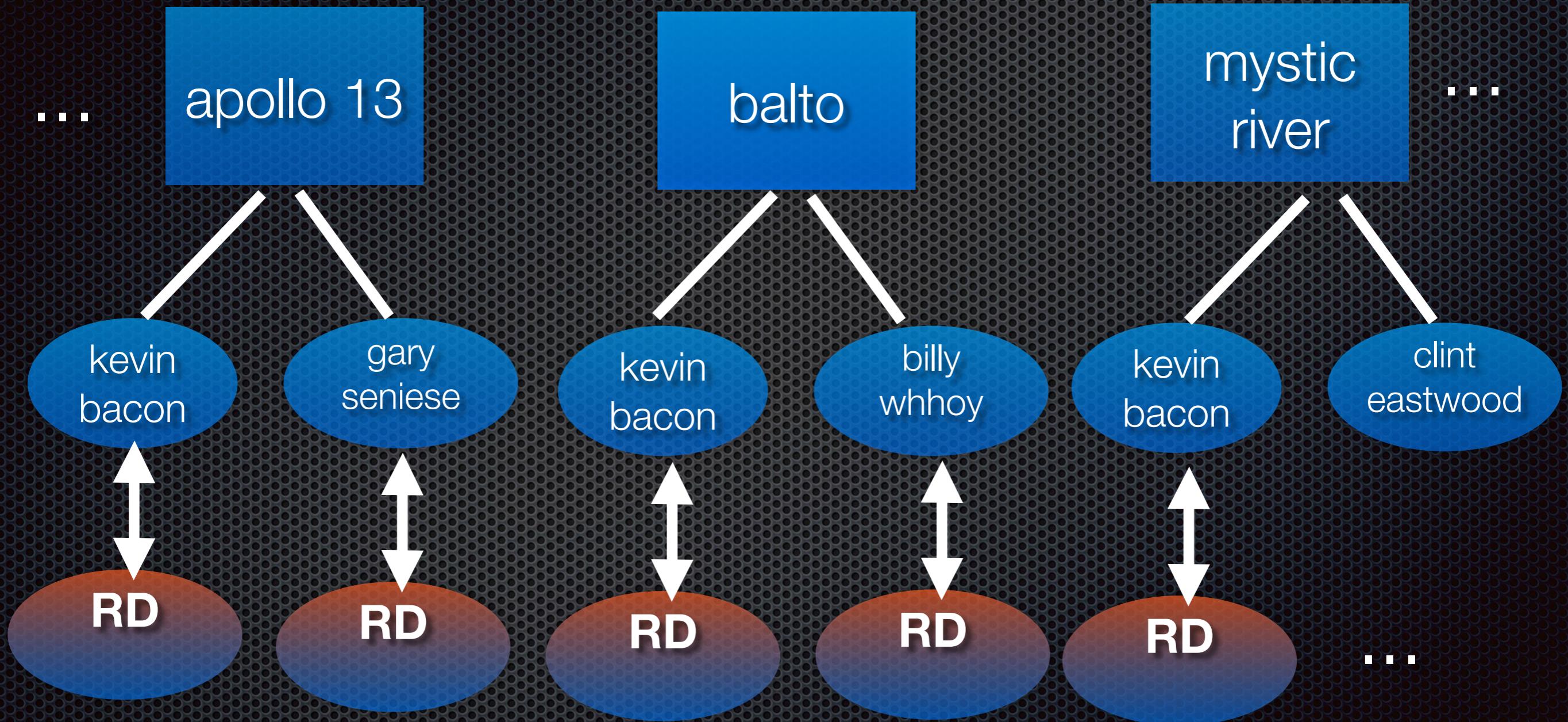
# Six Degrees: Algorithm

- L0: find all the films that KB is in
  - is RD a co-star in any of those films ?
    - If yes, stop and say degree is 0
    - 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....

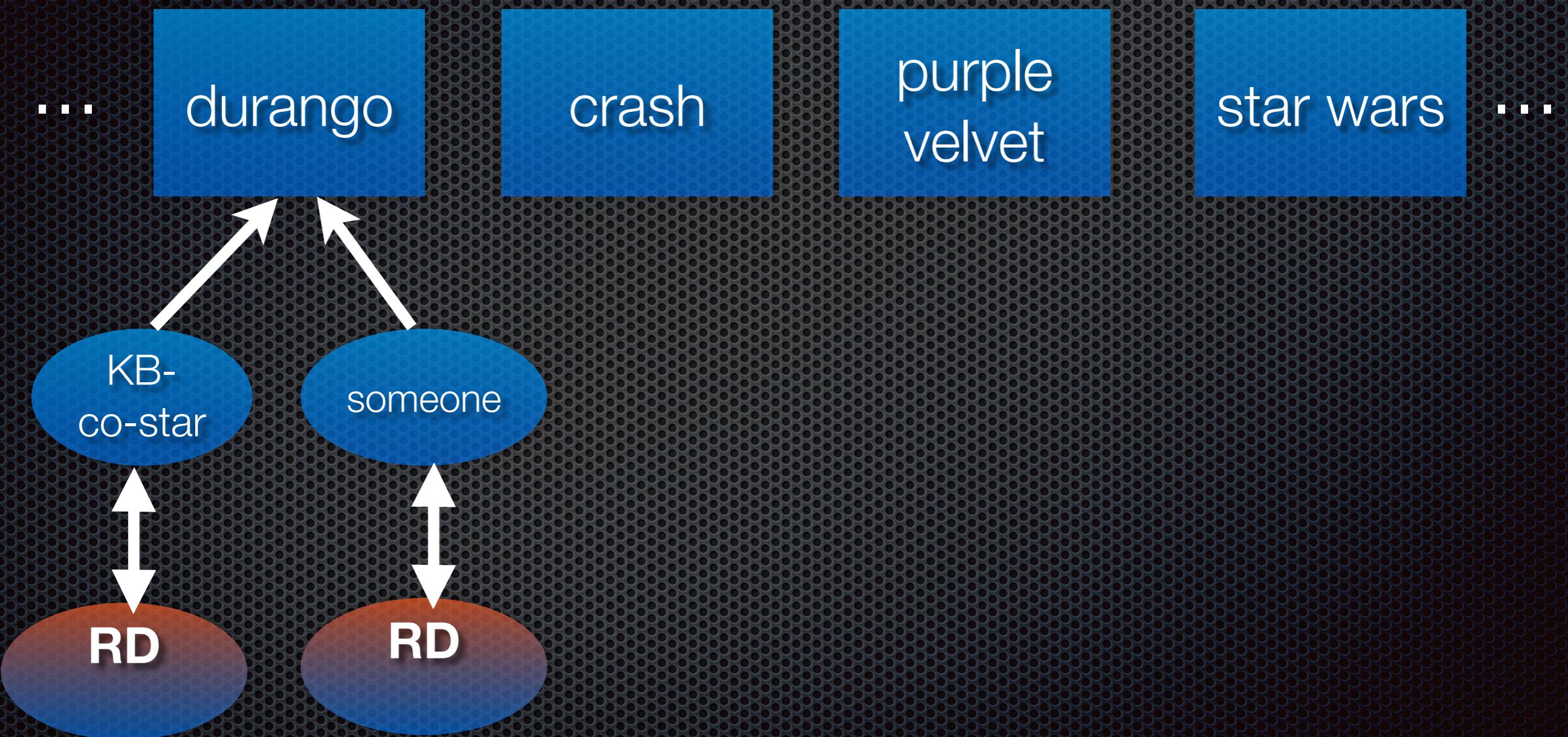
Kevin Bacon (KB)

Richard Dreyfuss (RD)

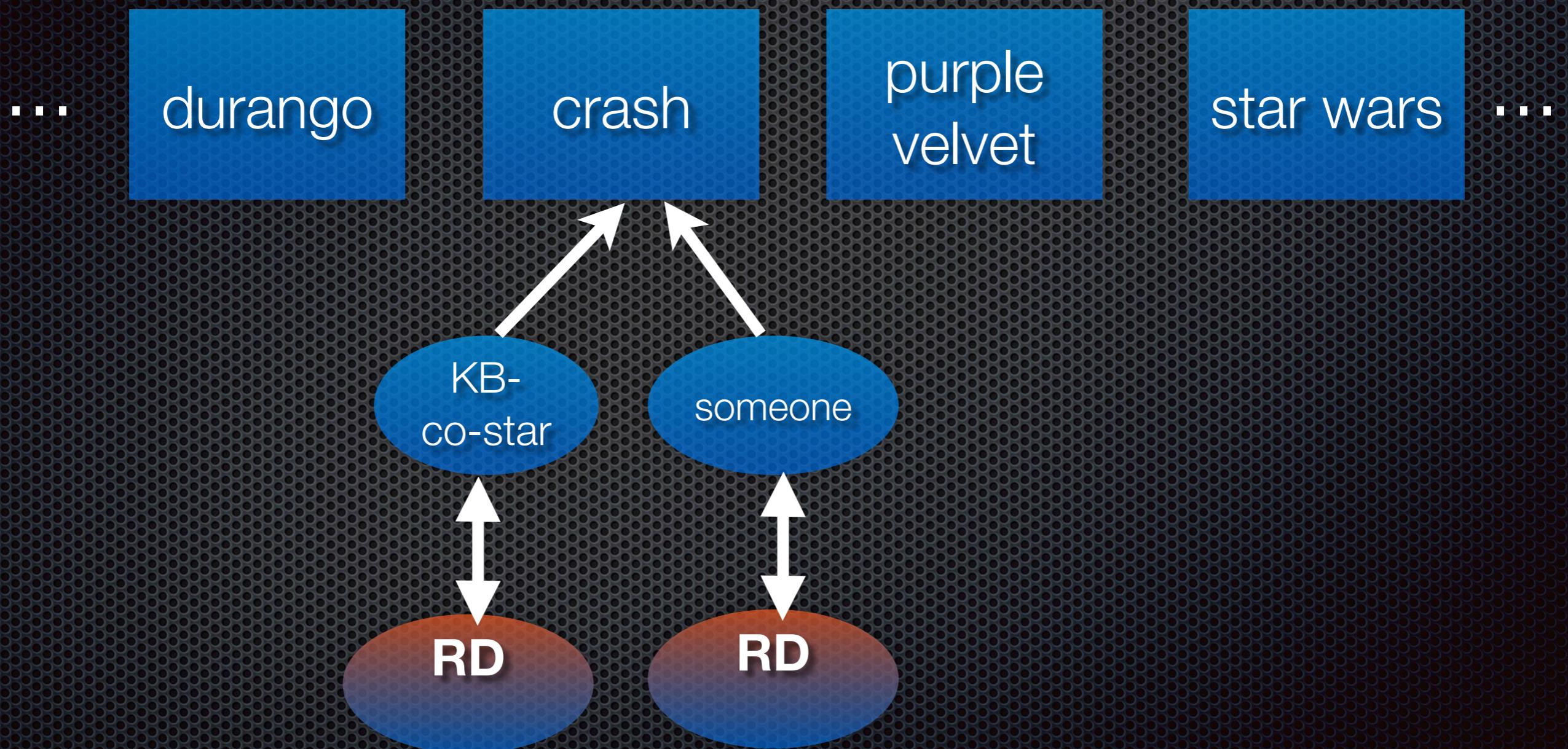
# L0 test: is RD a co-star of KB this film?



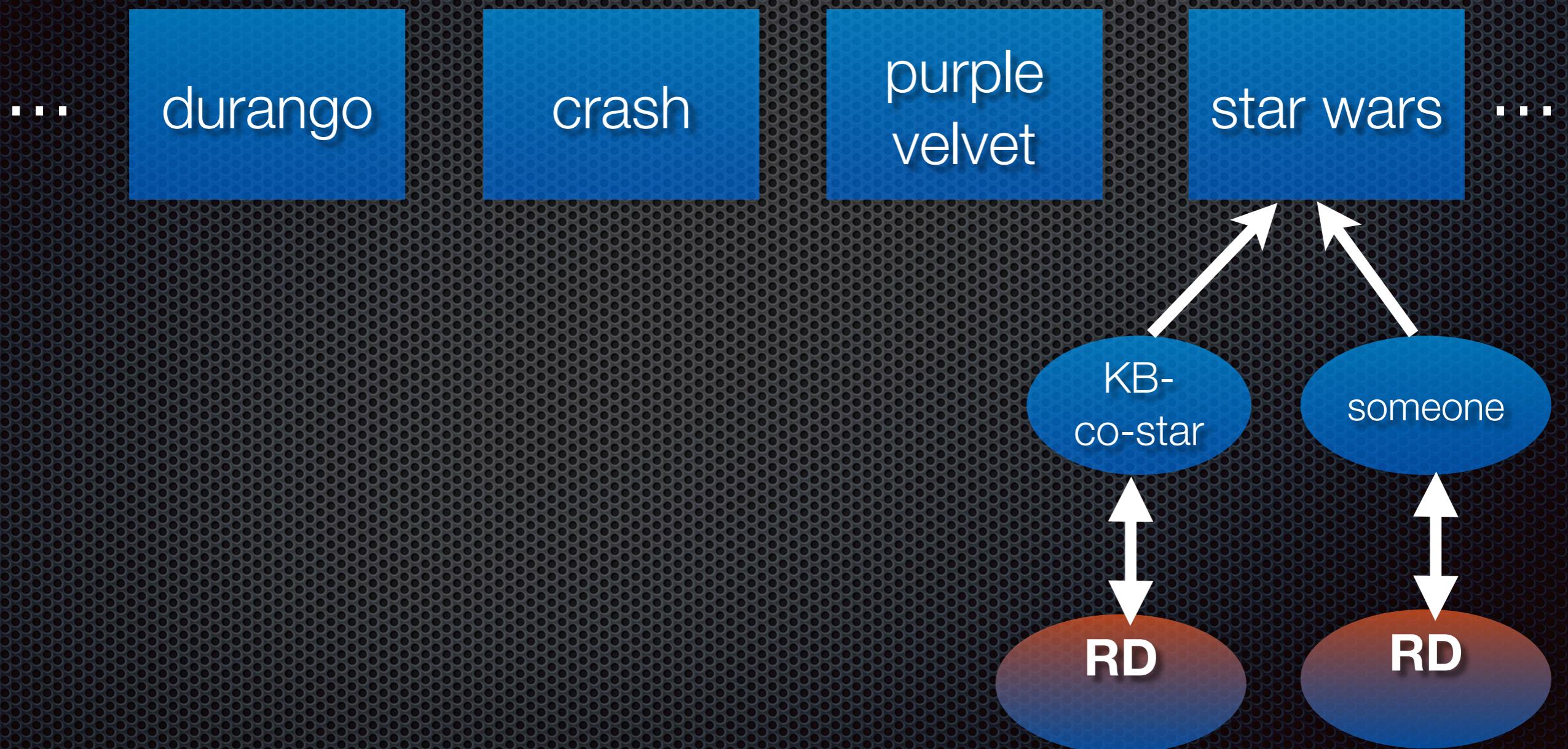
Ln test: is RD a co-star of KB-co-star from previous level ?



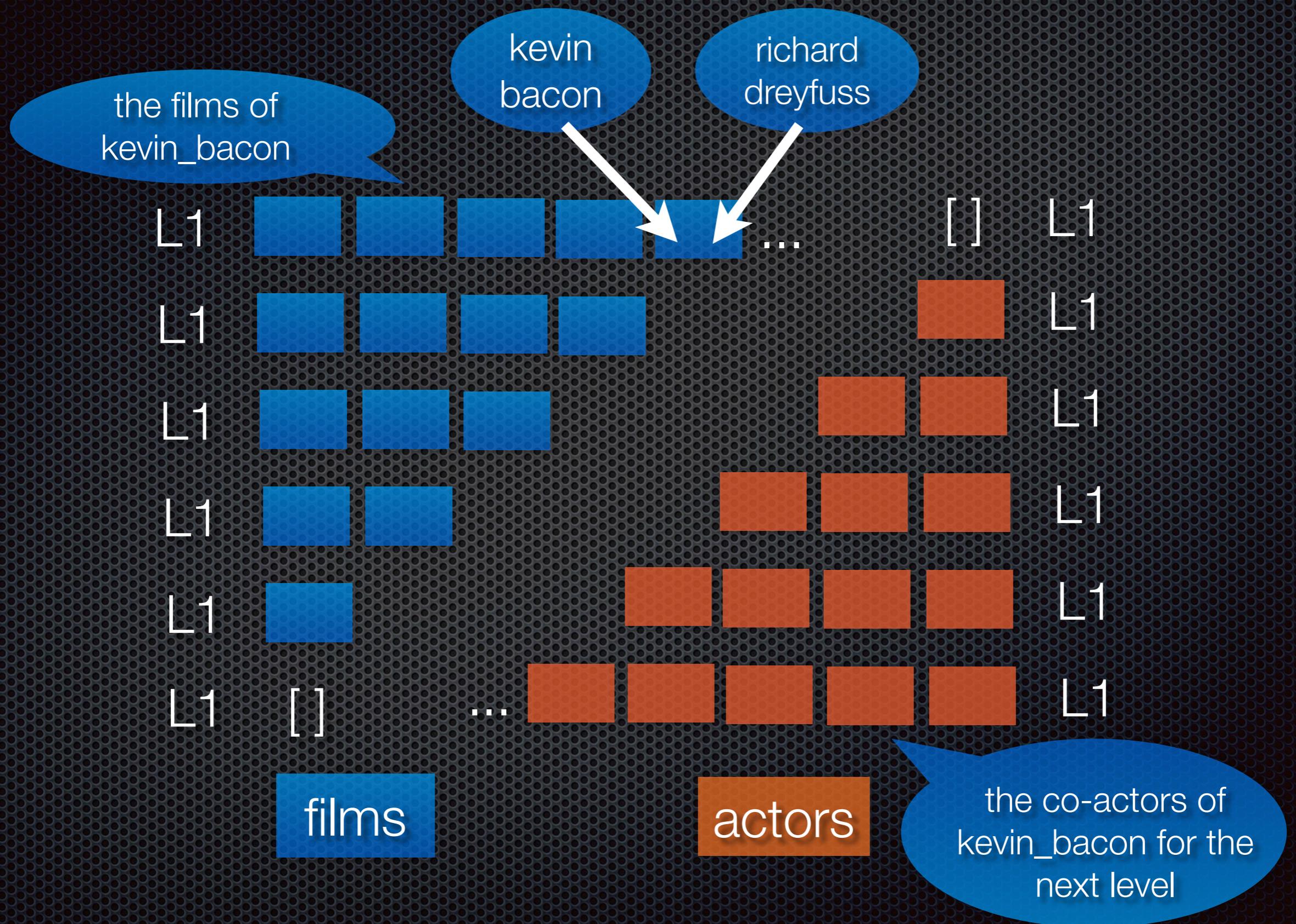
Ln test: is RD a co-star of KB-co-star from previous level ?



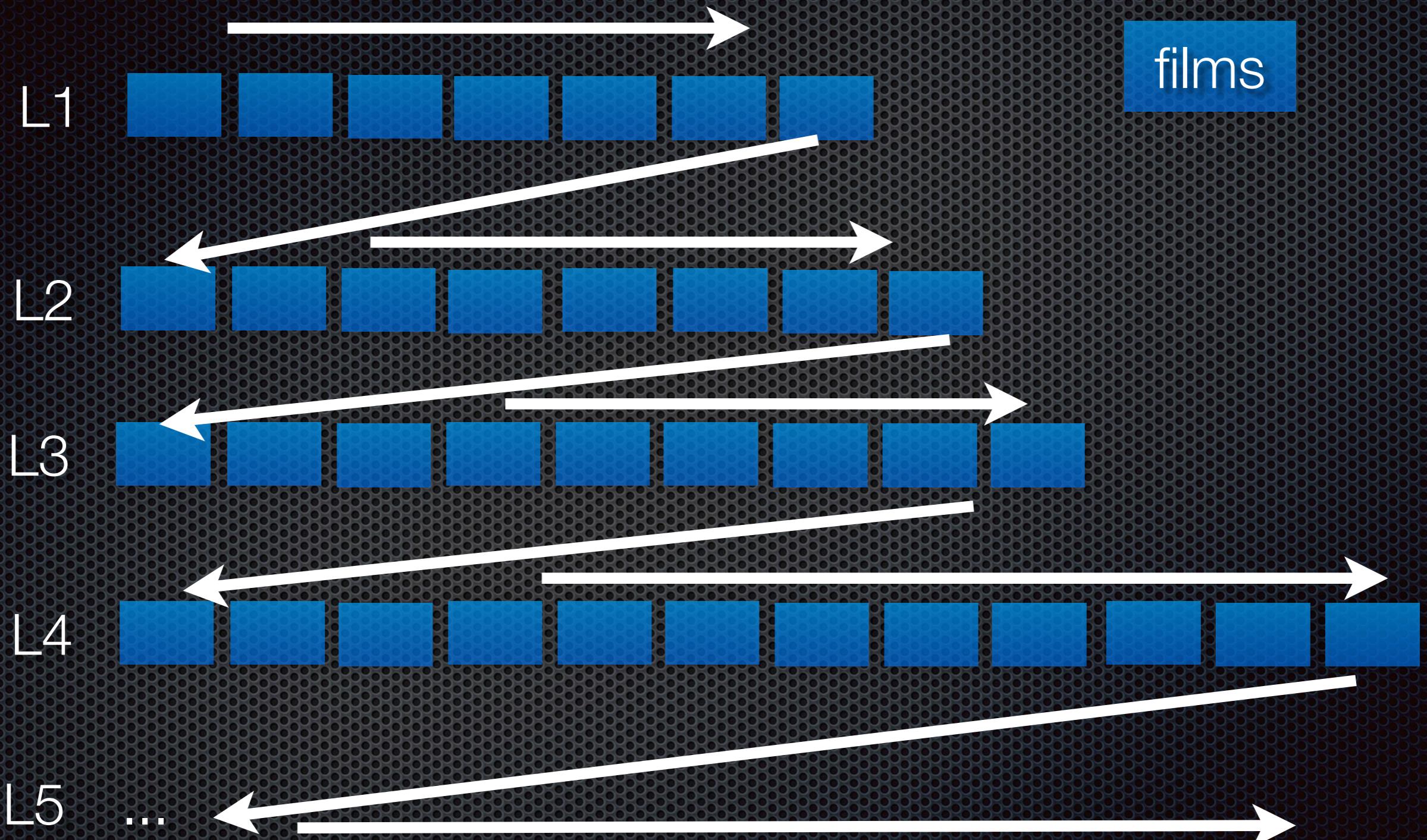
Ln test: is RD a co-star of KB-co-star from previous level ?



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



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



## 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

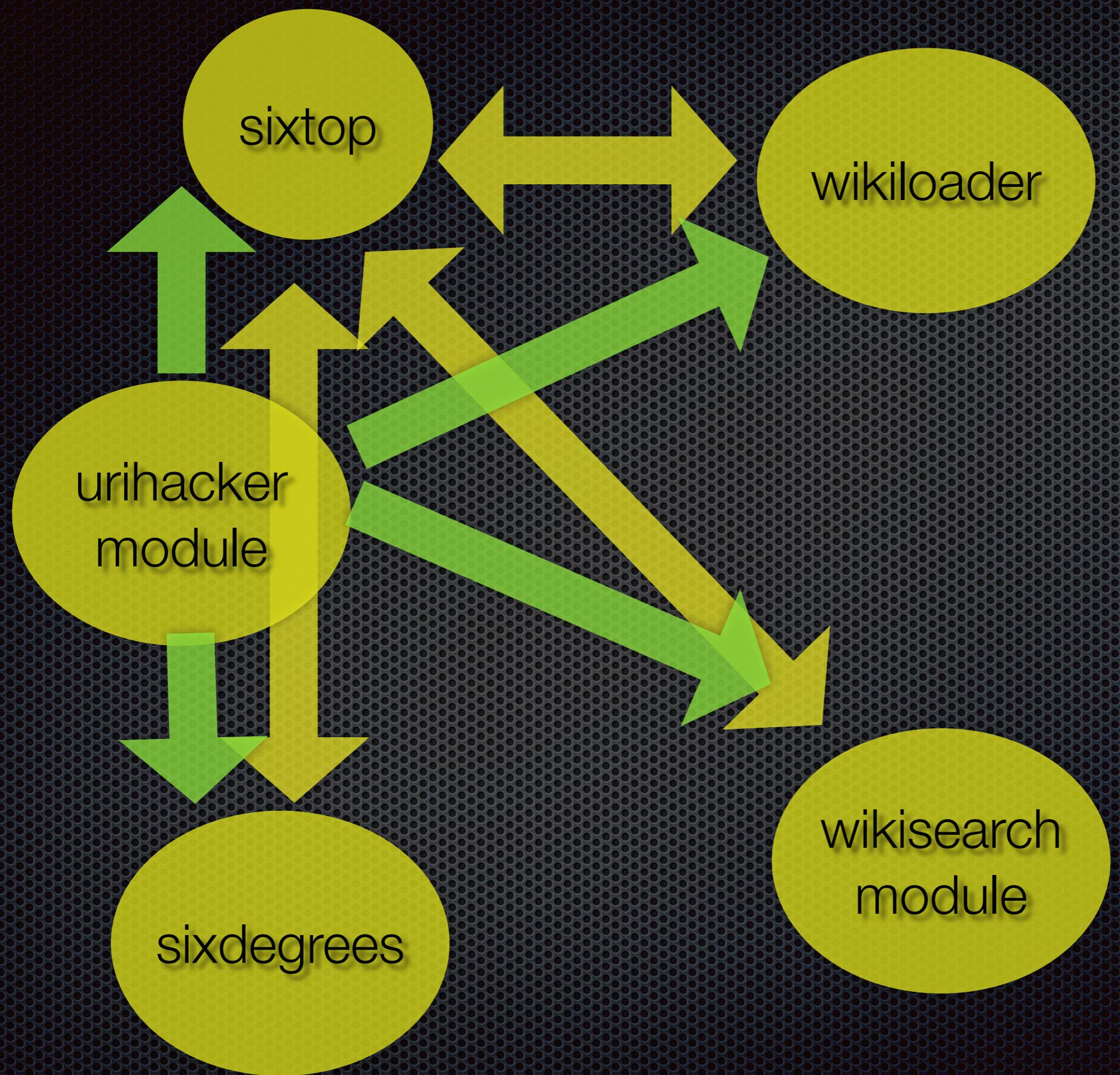
# Six Degrees Implement

- 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](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 !



# 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
```

# WikiLoader v.1

```
def load_uri(uri, flag)
  puts "\nChecking the following: #{strip_out_name(uri)}"
  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
```

# 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
```

# 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 "Error, 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 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
```

*sixton rh*

```

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
    ...
  end
end

```

*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"
  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
    ...
  end
end
```

*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"
    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
      ...
    end
  end
end

```

cond2: both lists empty

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
  end
end

```

*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
```

cond3: only the base\_actor is left  
and film list is empty

```
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)}}
```

action3: move to next level

```
  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
  check_level
cond4: some actors but no films
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
action4: move along this level

```

*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
```

*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
```

cond5: no actors, but some films

action5: get some actors

*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
```

test5 is used the most  
sixdegrees.rb

action6: pop a film and get its actors

cond6: some actors, some films

```
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
```

# 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

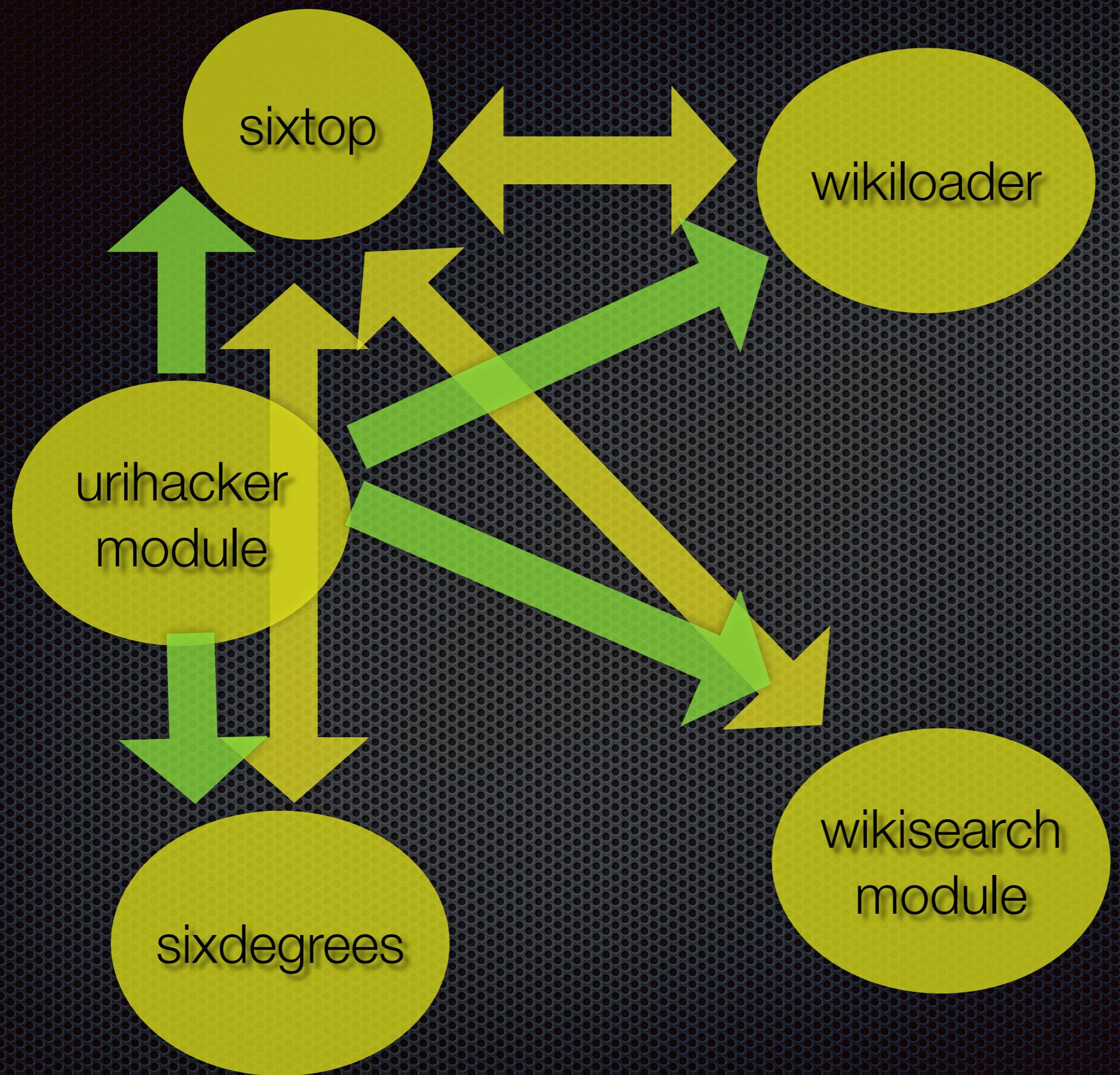
**strip\_out\_name**'s  
problems with **nil** and **[]**

*sixtop.rb*

# Other things in sixtop.rb

```
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*



Part F:  
Six degrees of separation...running

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

*sixton rh*

Checking the following: [http://en.wikipedia.org/wiki/Kevin\\_Bacon](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\\_\(1984\\_film\)](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_Hessemann",  
"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\_Hessemann", "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\\_\(2011\\_film\)](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\\_\(film\)](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/Illeana_Douglas)"

"[http://en.wikipedia.org/wiki/Richard\\_Dreyfuss](http://en.wikipedia.org/wiki/Richard_Dreyfuss)"

"Using test5"

Checking the following: [http://en.wikipedia.org/wiki/Lansky\\_\(film\)](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/Richard_Dreyfuss)", "[http://en.wikipedia.org/wiki/Eric\\_Roberts](http://en.wikipedia.org/wiki/Eric_Roberts)", "[http://en.wikipedia.org/wiki/Anthony\\_LaPaglia](http://en.wikipedia.org/wiki/Anthony_LaPaglia)"]

"[http://en.wikipedia.org/wiki/Illeana\\_Douglas](http://en.wikipedia.org/wiki/Illeana_Douglas)"

"[http://en.wikipedia.org/wiki/Richard\\_Dreyfuss](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](mailto: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...**