Ruby
Practical 7
Student Name- Vipin Hans
Student Number- 17201230

1) In the SixDegrees program, the **find_films_by_actor** method is fairly crap. If you look at any web-page it clearly misses loads of the films that the actor is in. Your task is to modify it to find more films (n.b., everything it finds should be a filmlink, dealing with garbage would cause too many problems in loading).

Answer-

I've made a few a few changes to the program to find out all films based on an actor. Please find my list of changes below

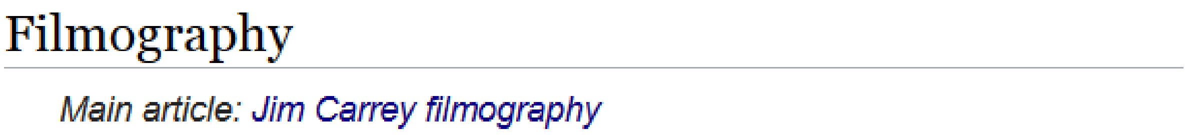
First of all, I'm accepting a film actor and invoking class instance of Sixdegrees

sixo = SixDegrees.new("Jim_Carrey", "Michael_Jackson")
sixo.check_levels

For efficient search and pulling all names of films.
Instead of navigating to below url
https://en.wikipedia.org/wiki/Jim_Carrey

I am navigating to
https://en.wikipedia.org/wiki/Jim_Carrey_filmography

There is also a link provided in every actors wikipage



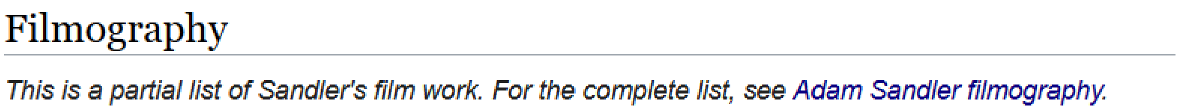
Filmography

Main article: Jim Carrey filmography

Which navigates to a filmography page.
https://en.wikipedia.org/wiki/Jim_Carrey_filmography

Another example would be

https://en.wikipedia.org/wiki/Adam_Sandler_filmography



Filmography

This is a partial list of Sandler's film work. For the complete list, see Adam Sandler filmography.

This will take us to Filmography page which lists all movies and series done by actor listed in a table.

To fetch the list, I've made below changes to the program.

Filename- urihacker.rb

To open the filmography page directly by sending actor name as parameter I've defined the following function-

```
def add_full_uri_actor
                  "http://en.wikipedia.org/wiki/" + self+"_filmography"

 end
```

Which is similar to original function

```
def add_full_uri
                  "http://en.wikipedia.org/wiki/" + self

end
```

Except that it will only be called for actors and not for films page.

Filename- wikisearcher.rb

Instead of modifying the original method I've created another method listed below to find films by actors.

```
   def find_films_by_actor1(out = [])
           puts "     ***** Entering find_films_by_actor with #{@title}."
           all_tables = @doc.search("//tr/td[2]/i/a[@href]")
           puts "printing all movies done by actor"
           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.select {|newnode| puts
newnode.inner_text}
           end

                   all_tables.each do |link|
                   link_info = link['href']
                   out << link_info
                   end
                   out.uniq.collect {|link| link.strip_out_name}
         end
```

To call this new function I've edited below class-

Filename- 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, :allow_redirections => :safe))
                        search_obj = WikiSearcher.new(doc_parse)
                        new_actors = search_obj.find_films_by_actor1
                        p new_actors.uniq.compact
                        p new_actors.size
                        out.concat(new_actors)
        elsif load_ok_flag && flag == :film
                then doc_parse = Nokogiri::HTML(open(uri, :allow_redirections => :safe))
                        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
```

```
sixo = SixDegrees.new("Jim_Carrey", "Michael_Jackson")
sixo.check_level
```

Filename- sixdegrees.rb

There were a few changes made to this file to call the above method. The changes are listed below-

```
        def initialize(base, target)
                @origin = base
                @target = target.add_full_uri
                @base_actor = base.add_full_uri_actor
                @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
```

These were all the changes made to fetch all films for the actors.

2) The way sixdegrees.rb searches through the space of actors and films is ok, but probably not the most efficient. Can you propose two modifications to the program that might search in a different way. Just propose the algorithm, you don't have to implement it.

Answer-

Yes, there can be changes to program.

First change can be the one I implemented to fetch all films from filmography page of actors instead of main actor page.

This lists all films done by actors in a table and you can fetch all films through loops.

Another change can be storing data in hash tables in a key value pairs like initials of first letters of names of films and their full names. For each actors we can compare key values of their hash tables and if two initials of the film names matches we can compare the whole string names of films and find out if they worked together in a film or not.

The concept of hash tables is because its faster, indexed, flexible and needs much fewer comparisons to find out relations between 2 actors.