

# Introduction

You are participating in an experiment in software engineering, which aims to provide insights on code complexity and comprehension. The experiment consists of two programming assignments and a short survey and in total takes approximately 30 mins to complete.

**Note:** this is not a test of your skills – it is an experiment about the code itself. It is completely anonymous and no personal information is being collected.

We thank you for participating and hope you enjoy!

## General Information

Age \_\_\_\_\_

Gender \_\_\_\_\_

Program of study (Computer Science, Computer Engineering etc.)  
\_\_\_\_\_

Degree (B.Sc./M.Sc./Ph.D) \_\_\_\_\_

Year of study (1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup> ...) \_\_\_\_\_




## Part A - Experiment Description

Online message boards consist of a hierarchy of objects –


### 1. Forums, which consist of a list of threads

	Board	Last Post	Threads	Posts
	<b>Bone Disorders</b> (22 Viewing)	<a href="#">Steroid induced bone loss</a> by MSNik 05-16-2016 10:55 AM 	1,849	6,595

### 2. Threads, which consist of a list of posts

	Thread / Thread Starter	Rating	Last Post	Replies ▼	Views
	<a href="#">Basal joint surgical treatment..Effective???</a> (  1 2 3 4 5 ... Last Page) Biyak123		08-05-2010 07:51 AM by martna	<a href="#">202</a>	126,650

### 3. Posts – written by a single person, and published in a single thread that is part of some forum

10-26-2008, 06:11 PM <span style="float: right;">#1</span>	
<p><b>Biyak123</b> Senior Member (female)</p> <p>Join Date: Jun 2005 Posts: 138</p> 	<p><b>Basal joint surgical treatment..Effective???</b></p> <p>I am to have basal joint surgery on my left thumb in three weeks. (Osteoarthritis) I would like to know how effective surgery was for others and the time of recovery. I do have a pamphlet but would like the opinions of others. Thanks.</p> <p>Biya</p>

The posts in a single thread can span a few pages. Links to those pages appear incomplete with several links to the first few pages and a link to the last page of posts (see red box above). When writing a crawler that needs to collect all those pages urls, we want to avoid unnecessary URL calls – in the example above, we don't want to get into page **#5** just so we can find the link to page **#6**. Therefore, we wrote a method that is able to automatically generate a complete list of pages urls based on the partial list of links that appears on the page.

You are given the **"GetAllHrefs"** method that given a partial list of page urls, returns a complete list of pages urls.

However, sometimes the link to the first page of URLs either has a unique structure or is missing entirely, which causes our method to output a completed list but **without a link to the first page**.

### Input

- list of input urls
- list of regular expression patterns extracting page number from a url

### Output

- list of completed urls

### Example

Given the following three urls –

```
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-2.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-3.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-17.html
```

the function provided to you outputs the following urls –

```
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-2.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-3.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-4.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-5.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-6.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-7.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-8.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-9.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-10.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-11.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-12.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-13.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-14.html
http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-15.html
```

<http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-16.html>  
<http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-17.html>

**Note that the first page is missing.**

### *Your Task*

Your task is to add the needed code to the function in order for the first page to be included in the result – there is no need to execute/compile the code or debug anything. A main() function is included for your convenience. Utilities are available but the task can be completed without inspecting them. Hard-coded solutions like the one below are not allowed...

```
ans.add("http://www.healthboards.com/boards/headaches-migraines/263061-topamax-question-1.html");
```

```
public static void main(String[] args) throws Exception
{
    ArrayList<String> u = new ArrayList<>();
    urls.add("http://www.healthboards.com/boards/headaches-
migraines/263061-topamax-question-2.html");
    urls.add("http://www.healthboards.com/boards/headaches-
migraines/263061-topamax-question-3.html");
    urls.add("http://www.healthboards.com/boards/headaches-
migraines/263061-topamax-question-17.html");

    Pattern pat = Pattern.compile("(?<=-)\\d+");
    ArrayList<Pattern> p = new ArrayList<>();
    patLst.add(pat);
    ArrayList<String> ans = GetAllHrefs(u, p);

    for (String a : ans)
        System.out.println(a);
}
```

```

public static ArrayList<String> GetAllHrefs(ArrayList<String> u,
ArrayList<Pattern> p)
{
    //if no urls, return empty list
    if (u == null || u.size() == 0)
        return new ArrayList<String>();

    String firstHref = u.get(0);
    if (u.size() < 3)
    {
        MyUtils.RemoveDuplicates(u, p);
        return u;
    }

    //get the longest common sequence (from the beginning of the
string) between each two adjacent urls
    String LCS = null;
    int maxLen = Integer.MIN_VALUE;
    for (int urlIndex = 0; urlIndex < u.size() - 1; urlIndex++)
    {
        String currLCS = MyUtils.GetAdjacentLCSStrings(u.get(urlIndex),
u.get(urlIndex+1), p, false);
        if (currLCS.length() > maxLen)
            LCS = currLCS;
    }

    if (LCS == null || LCS.isEmpty())
        return u;
    //get the page numbers using regular expressions
    ArrayList<Integer> n = MyUtils.getPageNumbers(u, LCS, 1, p);

    //get the longest common sequence (from the end of the string)
between each two adjacent urls
    ArrayList<String> ReversedReversedLCSS = MyUtils.GetReverseLCS(u,
p);
    String ReversedReversedLCS =
MyUtils.GetLongestString(ReversedReversedLCSS);

    //if too few page numbers, return
    if (n.size() < 3)
    {
        MyUtils.RemoveDuplicates(u, p);
        return u;
    }

    n = MyUtils.fillMissingNumbers(n);

    ArrayList<String> retUrls = MyUtils.ConstructUrls(n, LCS,
ReversedReversedLCS, firstHref, p);
    MyUtils.RemoveDuplicates(retUrls, p);
    return retUrls;
}

```