

Project: Web Crawler

For the purposes of this project, we define the Internet as the test data in this document, and a web crawler as software that requests pages from the Internet, parses the content to extract all the links in the page, and visits the links to crawl those pages, to an infinite depth.

Project Guidelines:

1. We do not expect candidates to spend more than 4 hours on this project.
2. Use any language you want.
3. Use any frameworks you want.
4. If you have to make a tradeoff between clean, maintainable code and a complete solution in the time you're able to spend, we would rather see clean code.
5. You may include a written statement containing:
 - a. Any assumptions you made
 - b. Reasons for any significant design choices you made
6. Your solution should:
 - a. Start with the first page in the list of pages and follow links to crawl the remaining pages in the list.
 - b. Visit each valid page in a JSON "Internet" exactly once. For example, if more than one page has a link to p2, you should only have to parse p2 one time.
 - c. Handle all the JSON "Internet" samples in the test data section.
 - d. Employ parallelism/concurrency if your language has those capabilities.
 - e. Return (order not important):
 - i. The list of page addresses it crawled successfully
 - ii. The list of duplicate page addresses it skipped
 - iii. The list of invalid page addresses it skipped

Test Data

Internet 1

```
{
  "pages": [
    {
      "address": "http://foo.bar.com/p1",
      "links": ["http://foo.bar.com/p2",
"http://foo.bar.com/p3", "http://foo.bar.com/p4"]
    },
    {
      "address": "http://foo.bar.com/p2",
      "links": ["http://foo.bar.com/p2",
"http://foo.bar.com/p4"]
    },
    {
      "address": "http://foo.bar.com/p4",
```

```

        "links": ["http://foo.bar.com/p5",
"http://foo.bar.com/p1", "http://foo.bar.com/p6"]
    },
    {
        "address": "http://foo.bar.com/p5",
        "links": []
    },
    {
        "address": "http://foo.bar.com/p6",
        "links": ["http://foo.bar.com/p7",
"http://foo.bar.com/p4", "http://foo.bar.com/p5"]
    }
]
}

```

Expected output

Success:

```

["http://foo.bar.com/p1", "http://foo.bar.com/p2",
"http://foo.bar.com/p4", "http://foo.bar.com/p5",
"http://foo.bar.com/p6"]

```

Skipped:

```

["http://foo.bar.com/p2",
"http://foo.bar.com/p4", "http://foo.bar.com/p1",
"http://foo.bar.com/p5"]

```

Error:

```

["http://foo.bar.com/p3", "http://foo.bar.com/p7"]

```

Internet 2

```

{
    "pages": [
        {
            "address": "http://foo.bar.com/p1",
            "links": ["http://foo.bar.com/p2"]
        },
        {
            "address": "http://foo.bar.com/p2",
            "links": ["http://foo.bar.com/p3"]
        },
        {
            "address": "http://foo.bar.com/p3",
            "links": ["http://foo.bar.com/p4"]
        },
        {
            "address": "http://foo.bar.com/p4",
            "links": ["http://foo.bar.com/p5"]
        }
    ]
}

```

```
    },
    {
      "address": "http://foo.bar.com/p5",
      "links": ["http://foo.bar.com/p1"]
    },
    {
      "address": "http://foo.bar.com/p6",
      "links": ["http://foo.bar.com/p1"]
    }
  ]
}
```

Expected output:

Success:
["http://foo.bar.com/p1", "http://foo.bar.com/p2",
"http://foo.bar.com/p3", "http://foo.bar.com/p4",
"http://foo.bar.com/p5"]

Skipped: ["http://foo.bar.com/p1"]

Error: []