1. Algorithms

1.1 Document Breakdown

1.1.1 Definition

Document breakdown is the way to break the document into many small parts. Each part has it own title and contents of it. And the final data has tree structure.

1.1.2 Define Problem

All content of document is quite difficute for manage so we must re-construc structure of document for using.

1.1.3 Solution

To solve this problem, we should follow these steps:

- Convert (save) document DOCX file as html type by using Microsoft Word save as Web Filtered.
- Import both html file and directory that incluses all pictures of document.
- Using xpath to get data of html file as we need, include h1, h2, h3,..., image, text content,..
- Save them with structure as below:
  -TitleA: contentA
  ---TitleA1: contentA1
  ------TitleA1.1: contentA1.1
  ------TitleA1.2: contentA1.2
  ---TitleA2: contentA2

1.1.4 Complexity

- In total, the complexity of this algorithm is N^4.

1.1.5 Flowchart

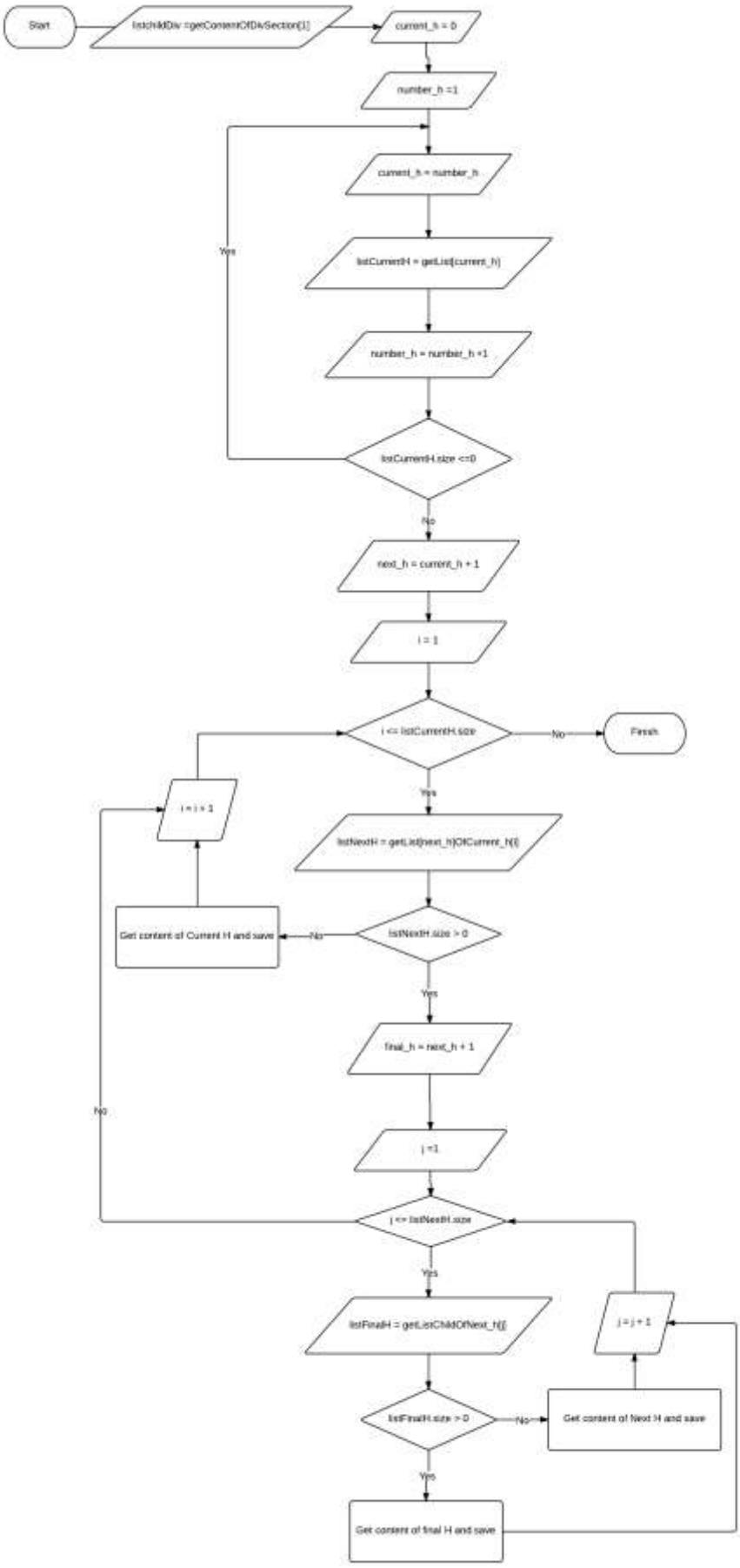

Start
listchildDiv =getContentOfDivSection[1]
current_h = 0
number_h =1
current_h = number_h
listCurrentH = getList[current_h]
number_h = number_h +1
listCurrentH.size <=0
Yes
No
next_h = current_h + 1
i = 1
i <= listCurrentH.size
No
Finish
Yes
i = i + 1
listNextH = getList[next_h]OfCurrent_h[i]
Get content of Current H and save
No
listNextH.size > 0
Yes
final_h = next_h + 1
No
j =1
j <= listNextH.size
Yes
listFinalH = getListChildOfNext_h[j]
j = j + 1
listFinalH.size > 0
No
Get content of Next H and save
Yes
Get content of final H and save

Figure 1: Breakdown document flow chart

1.2 Array Comparison

1.2.1 Define Problem

Given 2 Arrays. Calculate their matching percent to get how many percent of the same elements.

1.2.2 Requirement

- Language independence: Work only for defined skills in database.
- Two array: s1 and s2
- Example:
  - s1 = [a1,b1,c1,d1]
  - s2 = [a2,b2,c1,d2,e2]

1.2.3 Solution

- For each array, we find out how many pairs are contained in it.
- Then, we use below formula to calculate matching percent.

$$similarity(s1, s2) = \frac{2 \times size(|s1 \cap s2|)}{|size(s1)| + |size(s2)|}$$

1.2.4 Example

Calculate the matching percentage of 2 arrays:

1- [MySQL,Java Servlet,OSX,Javascript,Web Service,JSP,XML]

2- [java platform,SQL server,Java Servlet,microsoft windows,Java,enterprise javabean,Javascript,Web Service, Linux,JSP,microsoft sql server,XML,C#,automation test]

- Find array of the same elements:
  [Java Servlet, Javascript, Web Service, JSP, XML]
- Calculate its matching percent.

$$similarity\left(\begin{Bmatrix}\{\text{MySQL, Java Servlet, OSX, Javascript, Web Service, JSP, XML}\},\\ \begin{Bmatrix}\text{java platform, SQL server, Java Servlet, microsoft windows,}\\ \text{Java, enterprise javabean, Javascript, Web Service, Linux,}\\ \text{JSP, microsoft sql server, XML, C\#, automation test}\end{Bmatrix}\end{Bmatrix}\right)$$

$$= \frac{2 \times |\{\text{Java Servlet, Javascript, Web Service, JSP, XML}\}|}{|\{\text{MySQL, Java Servlet, OSX, Javascript, Web Service, JSP, XML}\}|} + \left|\begin{Bmatrix}\text{java platform, SQL server, Java Servlet, microsoft windows,}\\ \text{Java, enterprise javabean, Javascript, Web Service, Linux,}\\ \text{JSP, microsoft sql server, XML, C\#, automation test}\end{Bmatrix}\right|$$

$$= \frac{2 \times 5}{7 + 14} = \frac{10}{21} = 0.47$$

1.3 Rabin-Karp Algorithm

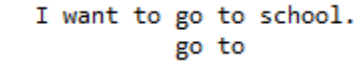
1.3.1 Define Problem

Given a document and a sentence, find out this sentence is existeds or not in this document.

1.3.2 Requirement

- After find out, give location of sentence which is exists in Document.
- Language: Multi language.

1.3.3 Example

- Two Strings:
  - String 1: I want to go to school.
  - String 2: go to
- Result:
  - 

. And give Location: 10 in String 1.