S184 Bogo Translate

Author: Alex Tung
Speaker: Steven Lau

January 27, 2018
Statistics

- Subtask 1: 40 contestants solved
- Subtask 2: 37 contestants solved
- Subtask 3: 36 contestants solved
- Subtask 4: 35 contestants solved

#contestants solved
**Problem Statement**

\[ N = 3 \text{ entries in translation database} \]

\[
\begin{array}{ll}
\text{WordA} & \text{WordB} \\
\hline
\text{charlie} & \text{charli} \\
\text{i} & \text{watashiwa} \\
\text{am} & \text{desu} \\
\end{array}
\]

\[ M = 1 \text{ translation task} \]

\[
\begin{array}{ll}
\text{sentence:} & \text{i am charlie} \\
\text{PattA:} & \text{SVO} \\
\text{PattB:} & \text{SOV} \\
\text{word-by-word:} & \text{watashiwa desu charli} \\
\text{final answer:} & \text{watashiwa charli desu} \\
\end{array}
\]

- \( 0 \leq N \leq 300 \), \( 1 \leq M \leq 10000 \)
- total \#words \leq 10000
- 1 to 26 words per sentence, 1 to 15 characters per word
- Subtask 1: Every word contains only one character
- Subtask 2: PattA = PattB
- Subtask 3: Empty translation database
- Subtask 4: No additional constraints
Subtask 1: Every word contains only one character

With word being a character, we can store translation database as a character array.

<table>
<thead>
<tr>
<th>WordA</th>
<th>WordB</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>m</td>
</tr>
<tr>
<td>r</td>
<td>f</td>
</tr>
<tr>
<td>m</td>
<td>s</td>
</tr>
</tbody>
</table>

```c
char WordA2WordB[128];
WordA2WordB[\'d\'] = \'m\';
WordA2WordB[\'r\'] = \'f\';
WordA2WordB[\'m\'] = \'s\';
```

Time complexity: $O(N)$
Subtask 1: Every word contains only one character

Word-by-word translation - with a word being a character, we don’t really need to break sentence into words.

```c
char sentence[1000];
gets(sentence);
int len = strlen(sentence);
for (int i = 0; i < len; i += 2)
    sentence[i] = WordA2WordB[sentence[i]];
```

Time complexity: $O(|\text{sentence}|)$
Subtask 1: Every word contains only one character

Pattern translation - build reverse lookup table

<table>
<thead>
<tr>
<th>PattA</th>
<th>PattB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVO</td>
<td>SOV</td>
</tr>
</tbody>
</table>

```c
char PattA[16], PattB[16];
scanf("%s%s", PattA, PattB);
//strlen(PattA) == strlen(PattB) == (len + 1) / 2
int r[128];
for (int i = 0; i < len; i += 2)
    r[PattA[i / 2]] = i / 2;
//r[\'S\'] = 0;
//r[\'V\'] = 1;
//r[\'0\'] = 2;
```
Subtask 1: Every word contains only one character

Pattern translation - build reverse lookup table

<table>
<thead>
<tr>
<th>PattA</th>
<th>PattB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVO</td>
<td>SOV</td>
</tr>
</tbody>
</table>

//r['S'] = 0;
//r['V'] = 1;
//r['O'] = 2;

char answer[1000];
strcpy(answer, sentence);
for (int i = 0; i < len; i += 2)
    answer[i] = sentence[r[PattB[i / 2]] * 2];

Time complexity: $O(|\text{sentence}|)$
Overall: $O(N + M \times |\text{sentence}|)$
Subtask 2: PattA = PattB

Cannot store translation database as character array.

<table>
<thead>
<tr>
<th>WordA</th>
<th>WordB</th>
</tr>
</thead>
<tbody>
<tr>
<td>charlie</td>
<td>charli</td>
</tr>
<tr>
<td>i</td>
<td>watashiwa</td>
</tr>
<tr>
<td>am</td>
<td>desu</td>
</tr>
</tbody>
</table>

```c
char WordA2WordB[128];
WordA2WordB['charlie'] = 'charli';
WordA2WordB['i'] = 'watashiwa';
WordA2WordB['am'] = 'desu';
//Compilation error. Why?
```
**Subtask 2: PattA = PattB**

Store translation database as string arrays

<table>
<thead>
<tr>
<th>WordA</th>
<th>WordB</th>
</tr>
</thead>
<tbody>
<tr>
<td>charlie</td>
<td>charli</td>
</tr>
<tr>
<td>i</td>
<td>watashiwa</td>
</tr>
<tr>
<td>am</td>
<td>desu</td>
</tr>
</tbody>
</table>

```c
char WordA[300][16];
char WordB[300][16];
WordA[0] = "charlie";
WordB[0] = "charli";
WordA[1] = "i";
WordB[1] = "watashiwa";
```

Time complexity: $O(N \times |\text{word}|)$
This time, we have to break a line into words.

```c
char  sentence[1000];
char  word[26][16];
int   words = 0;
char *p = strtok(sentence, " ");
while (p != NULL) {
    strcpy(word[words], p);
    words++;
    p = strtok(NULL, " ");
}
```
Subtask 2: PattA = PattB

Word-by-word translation:

```c
for (int i = 0; i < words; i++)
    for (int j = 0; j < N; j++)
        if (strcmp(word[i], WordA[j]) == 0) {
            strcpy(word[i], WordB[j]);
            break;
        }
```

No need pattern translation

Time complexity: $O(\#\text{words} \times N \times |\text{word}|)$

Overall: $O(N \times |\text{word}| + \text{total } \#\text{words} \times N \times |\text{word}|)$
Subtask 2: $\text{PattA} = \text{PattB}$

What if we make use of C++ Standard Template Library?
Subtask 2: PattA = PattB (C++ STL)

Store translation database as string map.

<table>
<thead>
<tr>
<th>WordA</th>
<th>WordB</th>
</tr>
</thead>
<tbody>
<tr>
<td>charlie</td>
<td>charli</td>
</tr>
<tr>
<td>i</td>
<td>watashiwa</td>
</tr>
<tr>
<td>am</td>
<td>desu</td>
</tr>
</tbody>
</table>

map<string, string> WordA2WordB;
WordA2WordB["charlie"] = "charli";
WordA2WordB["i"] = "watashiwa";
WordA2WordB["am"] = "desu";

Time complexity: $O(|\text{word}| \times N \times \log N)$

Where does the $\log N$ come from? Attend Data Structures (II).
Subtask 2: PattA = PattB (C++ STL)

Use string stream to break sentence into words.

```cpp
string sentence;
getline(cin, sentence);
stringstream ss(sentence);
string word[26];
int words = 0;
while (ss >> word[words])
    words++;
```

Word-by-word translation:

```cpp
for (int i = 0; i < words; i++)
    if (WordA2WordB.count(word[i]))
        word[i] = WordA2WordB[word[i]];  
```

No need pattern translation
Time complexity: $O(\#\text{words} \times |\text{word}| \times \log N)$
Lesson learnt

- C++ is very powerful
- Attend training next week: Introduction to C++
### Subtask 3: Empty translation database

Pattern translation - build reverse lookup table

<table>
<thead>
<tr>
<th>PattA</th>
<th>PattB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVO</td>
<td>SOV</td>
</tr>
</tbody>
</table>

//r['S'] = 0;
//r['V'] = 1;
//r['O'] = 2;

string ordered_word[26];
for (int i = 0; i < words; i++)
  ordered_word[i] = word[r[PattB[i]]];

Time complexity: $O(\#\text{words} \times |\text{word}|)$
Subtask 4: No additional constraints

Just combine subtask 2 and subtask 3.

Questions?