Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8166814	04:16:08	Suffix Array Re-construction	✔ **Accepted**	1.10 s	C++
	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
suffixarrayreconstruction.cpp	2243 bytes	1997f7f35093b094366e07f253c60bede2a494c9	download

Edit and resubmit this submission.

suffixarrayreconstruction.cpp

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

struct Suffix {
    int position;
    std::string suffix;
};

const char PLACEHOLDER = '#';
const char WILDCARD = '*';

inline bool insert_suffix(std::string& target, Suffix& s) {
    bool wildcard = false;
    for (int suffix_pos = 0, target_pos = s.position;
         target_pos < target.size();
         suffix_pos++, target_pos++) {

        if (s.suffix[suffix_pos] == WILDCARD) {
            wildcard = true;
            break;
        }

        if (target[target_pos] != PLACEHOLDER &&
            target[target_pos] != s.suffix[suffix_pos])
            return false;
        else
            target[target_pos] = s.suffix[suffix_pos];
    }

    if (wildcard) {
        for (int suffix_pos = s.suffix.size() - 1, target_pos = target.size() - 1;
             s.suffix[suffix_pos] != WILDCARD;
             suffix_pos--, target_pos--) {

            if (target[target_pos] != PLACEHOLDER &&
                target[target_pos] != s.suffix[suffix_pos])
                return false;
            else
                target[target_pos] = s.suffix[suffix_pos];
        }
    }

    n true;
}

```

```
48
49 std::string solve(int length, std::vector<Suffix>& suffixes) {
50     std::string result;
51     result.resize(length, PLACEHOLDER);
52
53     for (Suffix& s : suffixes) {
54         if (!insert_suffix(result, s)) return "";
55     }
56
57     for (char c : result) {
58         if (c == PLACEHOLDER) return "";
59     }
60
61     return result;
62 }
63
64 int main() {
65     int test_count;
66     std::cin >> test_count;
67
68     while (test_count--) {
69         int length, suffix_count;
70         std::cin >> length >> suffix_count;
71
72         std::vector<Suffix> suffixes;
73         suffixes.reserve(suffix_count);
74
75         while (suffix_count--) {
76             Suffix s;
77             std::cin >> s.position >> s.suffix;
78             s.position--; // Make position zero indexed.
79             suffixes.push_back(s);
80         }
81
82         auto result = solve(length, suffixes);
83
84         if (result.empty()) printf("IMPOSSIBLE\n");
85         else printf("%s\n", result.c_str());
86     }
87 }
```