

# UVa: The Paths Less Taken

UVa388 - Galactic Import

UVa768 - Crossword (REALLY interesting, no solution yet)

UVa479 - Irrigation Flow Rates

UVa777 - Codebreakers [todo check]

UVa999 - Book Signatures ✓G (构造法-直接构造解)

UVa967 - Circular ✓G (筛法+暴力+打表)

UVa10060 - A hole to catch a man

UVa12557 - Detecting Code Snippets [doing]

UVa310 - L-system ✓△ (Curated BFS)

- Note: USE `getline(*)` instead of `cin!!` because `omega` and `z` can be blank strings.

- Input like:

```
while (!cin.eof()) {
    getline(cin, u);
    if (cin.eof()) break;
    getline(cin, v); getline(cin, o); getline(cin, z);
    if (u == "" || v == "") break;
```

- BFS + heavy pruning. Prune everything with `length > length of z`, so the upper bound

for our search is roughly  $2^{16}$  (time to generate all subsets). Code below:

```
int main()
{
    string u, v, o, z;
    while (!cin.eof()) {
        getline(cin, u);
        if (cin.eof()) break;
        getline(cin, v); getline(cin, o); getline(cin, z);
        if (u == "" || v == "") break;
        queue<string> q;
        unordered_set<string> visited;
        bool found = 0;
        long sz = z.size();
        for(long i = 0; i < o.size(); ++i) {
            string k = o.substr(i, sz);
            q.push(k); visited.insert(k);
        }
        while (!q.empty()) {
            string curr = q.front(); q.pop();
            string a, b, c;
            if (curr == z) {
                found = 1; break;
            }
            for(const char& ch : curr)
```

```

        switch (ch) {
            case 'a': c += u; break;
            case 'b': c += v; break;
        }
        for(long i = 0; i < c.size(); ++i) {
            string k = c.substr(i, sz);
            if (visited.find(k) == visited.end())
                q.push(k); visited.insert(k);
        }
    }
    if (found) printf("YES\n");
    else printf("NO\n");
}

```