

K-LOGCFL 解説

原案・解説 ynymxiaolongbao

問題概要

- 0以上N未満の値を取る整数変数 x
 - スタック s
 - 遷移にかかる係数の積 w
- を変数に持つプログラムを考える。

Initialize the following variables:

```
1 integer x = 0;  
2 integer w = 1;  
3 stack s = {};
```

問題概要

- ステップがN回ある。各ステップでは、パラメータ y を決めて、「スタックをpushする+ α 」か、「スタックをpopする+ α 」の処理をする。

If $0 \leq y_t$, do the following:

```
1 w *= A[t][x][y];  
2 s.push(x);  
3 x = y;
```

The above y denotes y_t .

If $y_t = -1$, do the following:

```
1 assert(!s.empty());  
2 w *= A[t][x][s.top()];  
3 x = (x + s.top()) % N;  
4 s.pop();
```

問題概要

- w は遷移方法に対する重みの積になっている。
- 任意の可能な遷移に対する w の和を求める。(つまり、任意の可能な遷移を漏れや重複がないように追ってくださいという問題)

If $0 \leq y_t$, do the following:

```
1 w *= A[t][x][y];  
2 s.push(x);  
3 x = y;
```

The above y denotes y_t .

If $y_t = -1$, do the following:

```
1 assert(!s.empty());  
2 w *= A[t][x][s.top()];  
3 x = (x + s.top()) % N;  
4 s.pop();
```

解法

- 以下のような、時刻に関する区間DPを組む $O(N^6)$
- $f(l, r, x_0, x_1)$... 時刻 l でpush, 時刻 $r - 1$ でその要素をpop
- $g(l, r, x_0, x_1)$... 時刻 l でpush, 時刻 $r - 1$ でそれと同じ深さの要素をpop(時刻 l の時点でのスタックの要素は途中で削られない)
- $h(l, x_0, x_1)$... 時刻 l の時点でのスタックの要素が途中で削られない
- ただし、 x_0 は直前の x の値、 x_1 は直後の x の値を表す

非決定性チューリングマシンと多項式時間DP

- いくつかの非決定性チューリングマシンが受理する言語のクラスは、Pの部分集合であることが動的計画法により示される。
- 今回は、メモリ $O(\log N)$ 、スタック一本の非決定性プログラム(の遷移に重みを付けたもの)だったので、LOGCFL

	メモリ	スタック	DP
正規言語	$O(1)$		耳DP
CFL	$O(1)$	1本	区間DP, 木DP
NL	$O(\log N)$		よくあるDP
LOGCFL	$O(\log N)$	1本	木DP, 区間DP

統計情報

- AC / 1 teams (凍結前)
- FA / Nya Revenge of shinchon (227 min)