



Problem E: Move On Dice

原案: 山口
解答: 宮村・山口
解説: 山口


問題概要

- + 各面に文字列の書かれた立方体がある
- + 正方形のマスのある並んだマップ上に置かれている
- + スタートからゴールまで立方体を動かせ
 - + 上面の文字列をつなぎあわせて辞書順最小となるように
- + 立方体を動かせる向きには制限があって、各マスに書かれている

解法概要 (要: サイコロライブラリ)

1. スタートとゴールから DFS or BFS をしてスタートから来れてゴールに行ける状態を列挙する
 - + このときもつ状態は {マップ上の位置} × {サイコロの向き}
2. 上の状態だけからなるグラフを作る
3. 1文字ずつ辞書順最小の方向に進む
 - + このときもつ状態は {マップ上の位置} × {サイコロの向き} × {何文字目}
 - + 辞書順最小の状態は1つでないので set でもつ
 - + このとき生成する文字列オブジェクトは全体で1つで良い
4. ゴールに着いたら文字列出力 / しきい値を超えたら "infinite"
 - + 基本的に "最強の呪文" と同じ
 - + 辺ではなく頂点に文字列が割り当てられているという点は異なる

サイコロライブラリ

- + マップ上に置かれたサイコロを転がしていく様子をシミュレートするライブラリ
 - + 操作: 上下左右に倒す・(時計回り or 反時計回りに横回転)
 - + 区別しないといけない状態: 24通り
 - +  × 6
 - + 状態遷移のテーブルを持っておく
 - + 24 (現状態) × 4 (操作) = 96 !?
 - + 少し考えればかなり減る
 - + 自分のライブラリの場合 16 個
- + 和が7になる数字の書き込み方はいくつもあるので展開図からの書き込み部分も用意しておくが良い
 - + 展開図上とマップ上では倒す方向が逆になるので注意！

Bellman-Ford による解法について

- + 文字列の辞書順について $\min(z + x, z + y) = z + \min(x, y)$ が成り立つことを利用した解法
 - + 非常に興味深い
 - + しかし重大な問題...
 - + AA...A ばかりを入れられると必要なループ数が増える上に共通接頭辞が非常に長くなって文字列比較にも時間がかかってしまう
 - + オーダーに響くレベル
 - + 今回の問題では通る場合があるが、次回からやめた方がおすすめ
- + もう1点:
 - + キーとして文字列を毎回生成しがちだが、メモリを浪費しすぎ
 - + 一般にこの手の探索/DP等では各ノードで隣への参照だけを持つ
 - + 遅いのは待てばなんとかなるときもあるが、メモリが足りないのはそもそも動かない

ゴールに着いたあと

- + 問題文には明記されていませんが、ゴールに着いたあとも立方体を動かし続けると考えても答えは同じ
 - + 接頭辞の方が辞書順で前に来るから
- + 注:
 - ⊂ 言語とその一派では終端記号 " ϵ " を最小元として扱うのでこうなる
 - 特定の論文では最大元として扱う場合もあり
 - そのときは異なる

おまけ

- + ジャッジ解
 - + 宮村 (C++): 280行くらい
 - + 山口 (Java): 240行くらい
- + Submit 状況
 - + Submit: 45 times
 - + Accepted: 4 teams

 - + First Accept: kaneko 176 min.
 - + First Accept (Onsite): wakaba