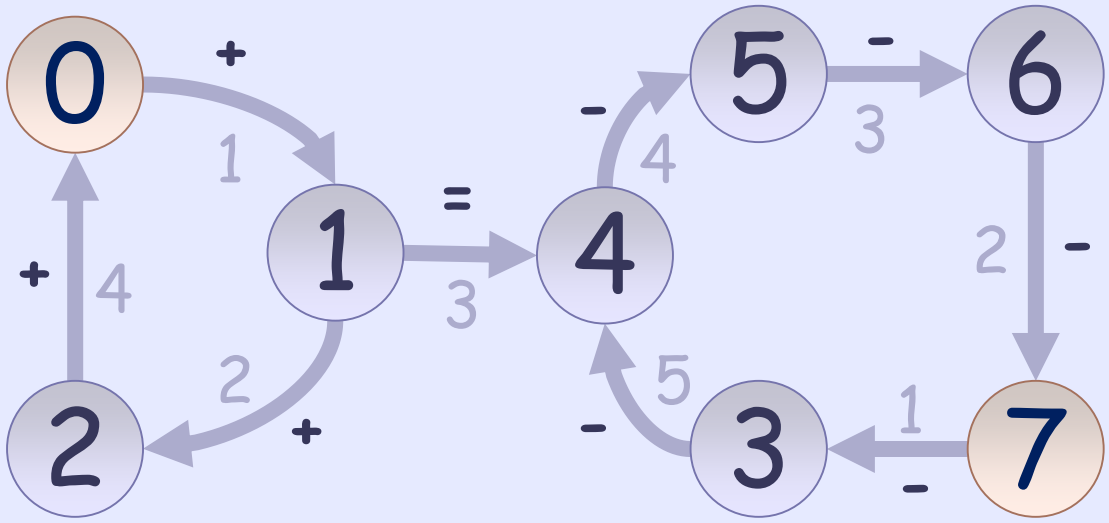
L: 1円ロード

問題案: 稲葉

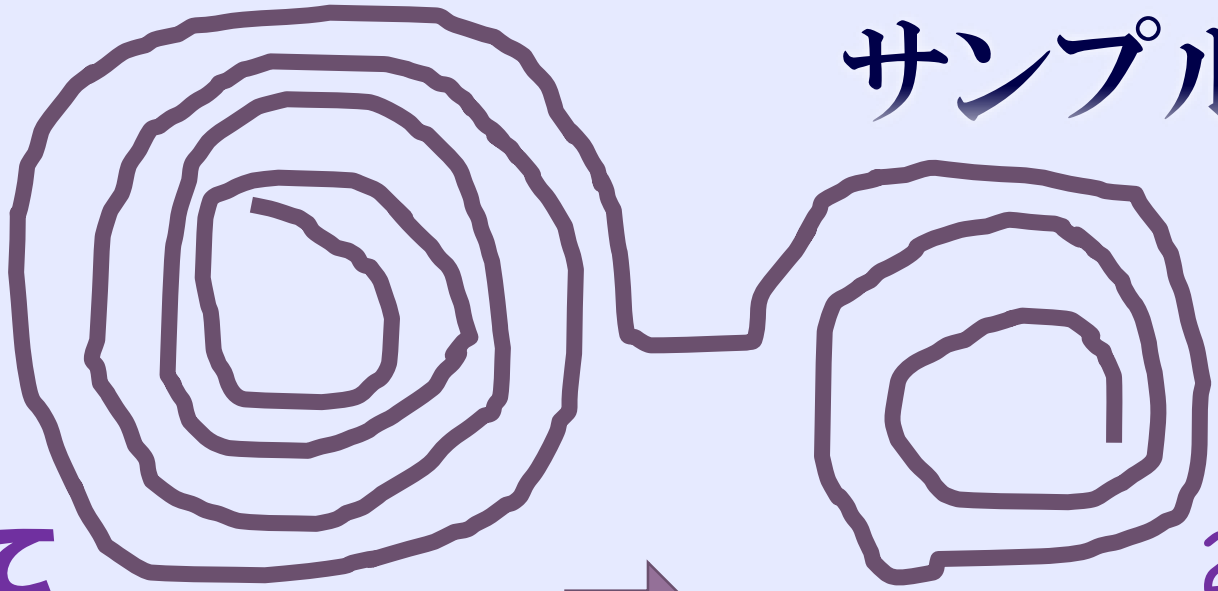
解答:秋葉、稲葉

問題

- 「+」の辺を通ると所持金が1円増える
- 「−」の辺を通ると1円減る（文無しは通れない）
- 始点を0円で出て終点に0円で着く最短路は？
  - |V| ≦ 250

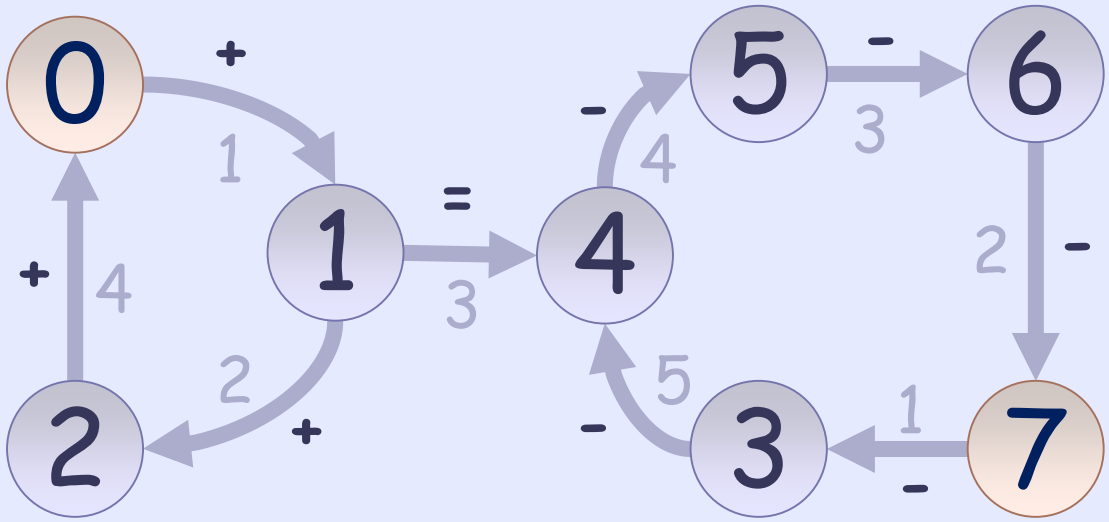


サンプルの解



4 周して
13=1+3*4 (= 3 mod 5)
円稼ぐ

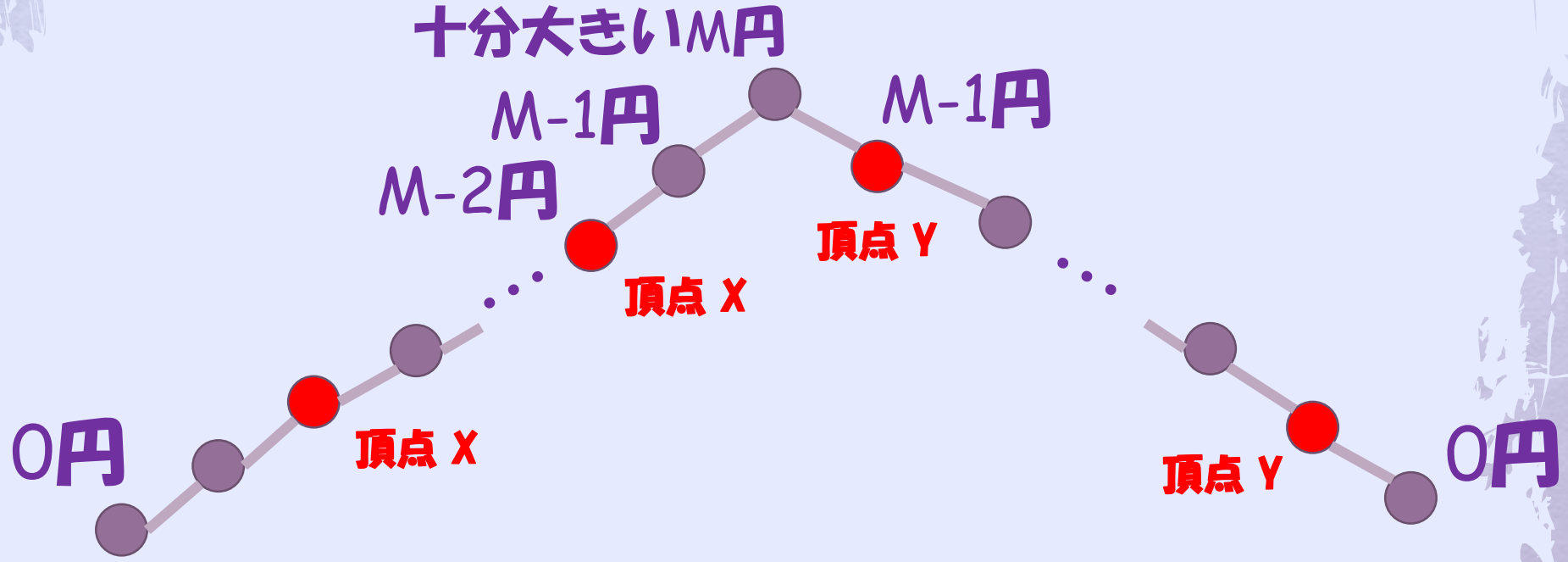
2 周して
13=3+5*2 円使う



部分点解法 O(|V|²|E|log|V|)

- 「所持金 × 頂点」の拡大グラフで Dijkstra
  - サンプルのように |V|² オーダの所持金は必要
  - 実はそれより多くは要らない

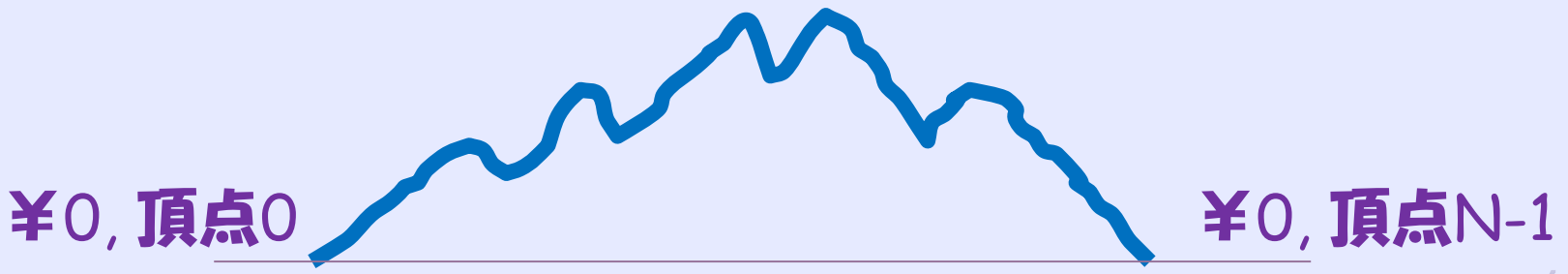
    [超略証] たくさん稼ぐ間にもたくさん使う間にも同じ頂点を何度も通るので、そういうサイクルをうまく相殺して消せる



満点解法

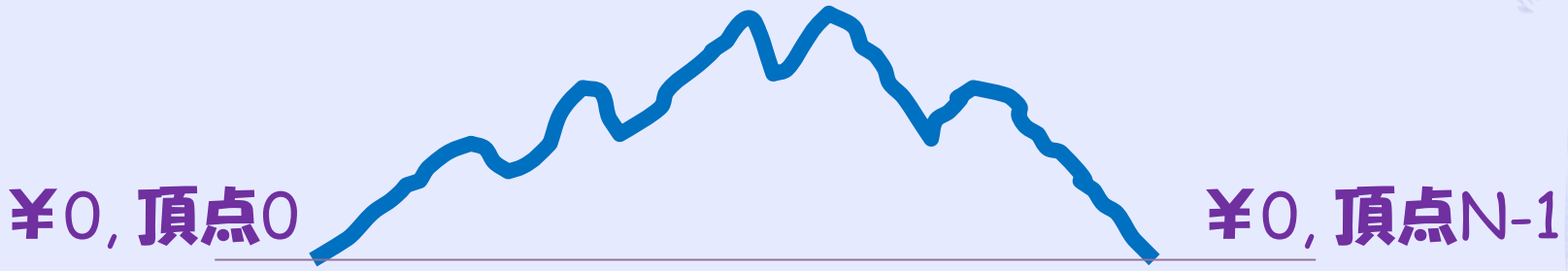
- 具体的な所持金額ではなく、
  始点から終点までの増減に着目

0円から0円までの最短路とは…

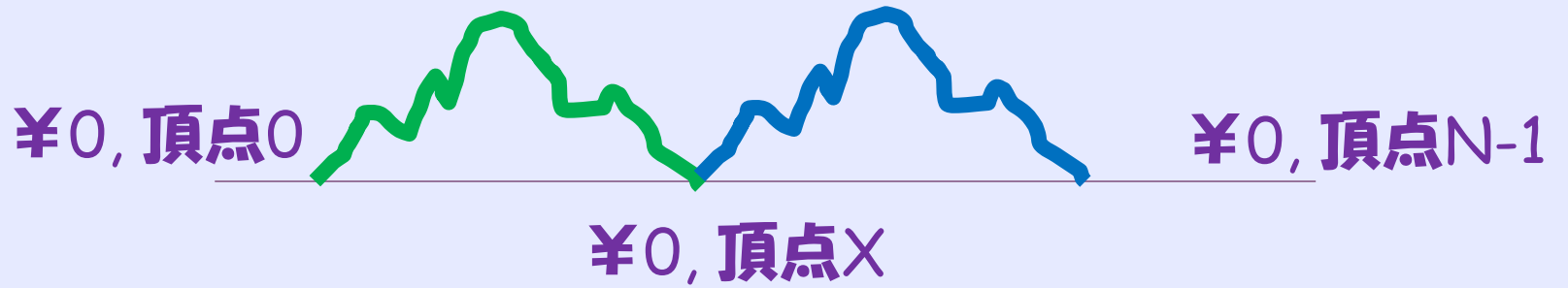


（※ 図は所持金増減チャート）

0円から0円までの最短路とは…

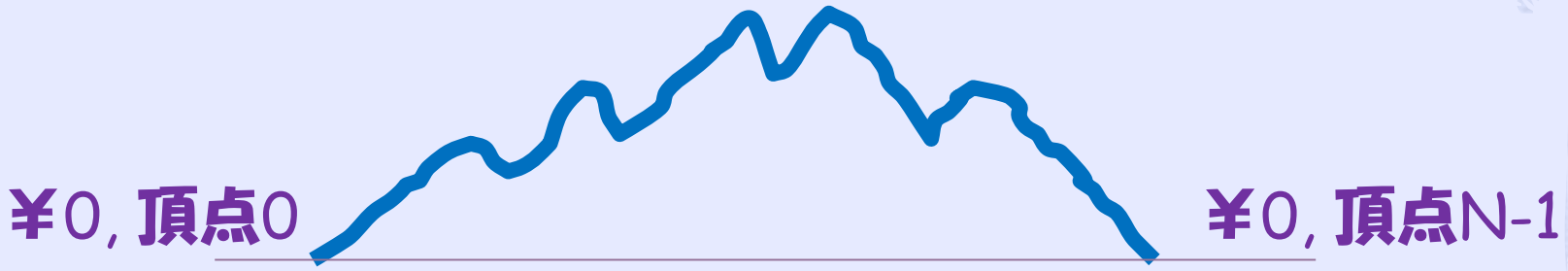


途中でどこかを0円で経由する最短路

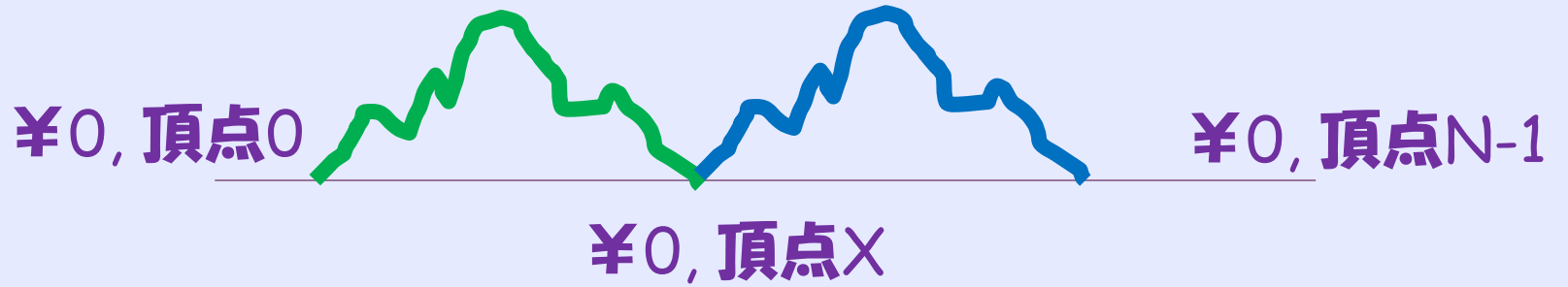


または…

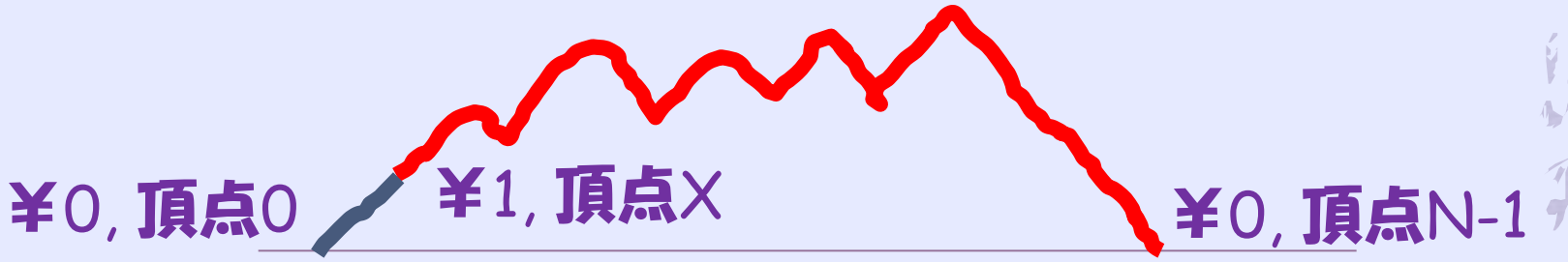
0円から0円までの最短路とは…



途中でどこかを0円で経由する最短路



または、すぐ1円になって最後に0円に戻る



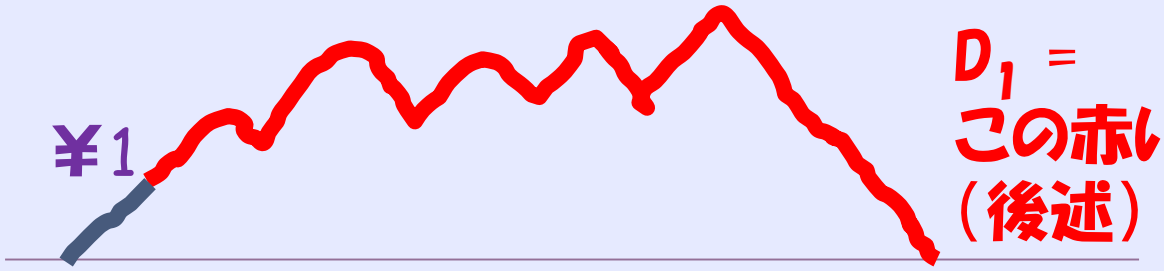
つまり

$D_0[a][b]$:= aからbまでの 0円 → 0円最短路

= min

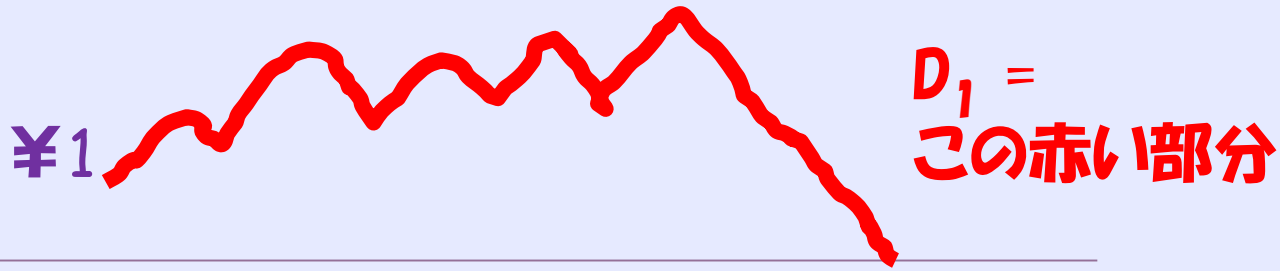
min_c { $D_0[a][c] + D_0[c][b]$ },

min_c { $L[a][c] + D_1[c][b]$ | $G[a][c]$= ‘+’ })



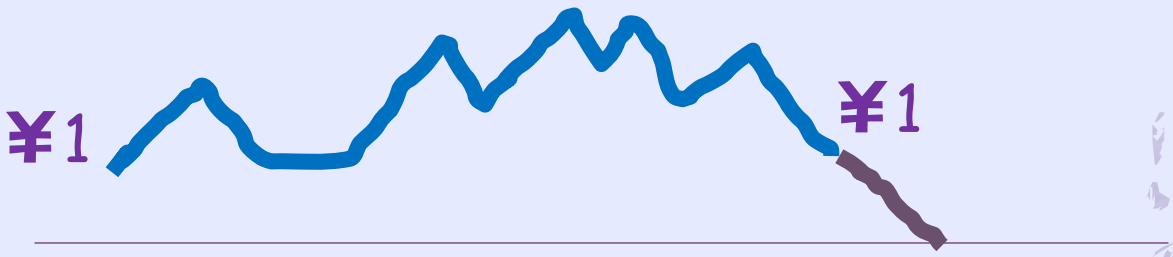
つまり（続き）

$D_1[a][b]$:= aからbまでの 1円 → 0円最短路



= min_c { $D_0[a][c]$ + $L[c][b]$ | $G[c][b]$= ‘−’ })

途中で0円に
落ちない
1円 → 1円
最短路
== 0円 → 0円
最短路！



満点解法

- というわけで
  - $D_0[][]$:= 0円 → 0円の全点間最短路
  - $D_1[][]$:= 1円 → 0円の全点間最短路

  の表を適宜埋めていけば$D_0[0][N-1]$が求まる

ただし…

満点解法

- というわけで
  - $D_0[][]$:= 0円 → 0円の全点間最短路
  - $D_1[][]$:= 1円 → 0円の全点間最短路

  の表を適宜埋めていけば$D_0[0][N-1]$が求まる

Warshall-Floyd的な
単純な３重ループは
間違い！！！

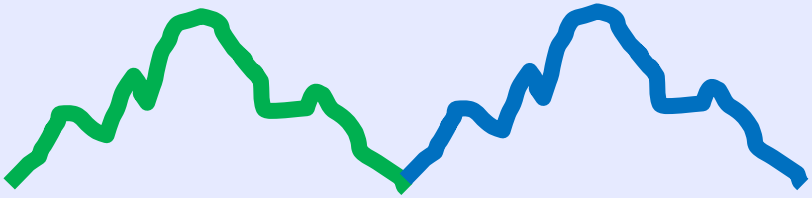
```
for (c)
  for (a)
    for (b) {
      D0[a][b] = min( ?[a][c] + ?[c][b] )
      D1[a][b] = min( ?[a][c] + ?[c][b] )
    }
```

満点解法

Warshall-Floyd的な
単純な３重ループは
間違い！！！

```
for (c)
  for (a)
    for (b) {
      D0[a][b] = min( ?[a][c] + ?[c][b] )
      D1[a][b] = min( ?[a][c] + ?[c][b] )
    }
```

- 普通の W-F は最大indexで最短路を2つに切れば(すでに前のループで求まってる)最短路
- 今回は所持金が 0 か 1 のところでしか切れないので、そこが最大indexの頂点とは限らない



満点解法 $O(|V|^3 \log |V|)$

Warshall-Floyd的な
単純な3重ループは
間違い！！！

```
for (c)
  for (a)
    for (b) {
      D0[a][b] = min( ?[a][c] + ?[c][b] )
      D1[a][b] = min( ?[a][c] + ?[c][b] )
    }
```

```
for ((dist,type,a,b) ∈ PriorityQueue)
  for (c)
    D[type][a][b] が埋まったので、
    その前後の D[a][c] と D[c][b] を埋めてみる
```

正解: PriorityQueueを使って、
短いマスから順に埋める

まとめ

- 頂点0 から 頂点N−1 の
  0→0円単一始点終点最短路のために
  0→0円と1→0円の全頂点間最短路を求める
- 全頂点間最短路なのに Priority Queue
- Keyword: “CFL Reachability”
  - この問題の到達可能性判定バージョンは、
    「関数を呼ぶ」を「+」、「return」を「−」として、
    プログラムの特定部分の影響がどこまで及ぶかの
    解析として、コンパイラなどで使われています