

日本橋ハーフマラソン 2026冬

(AtCoder Heuristic Contest 060)

“Ice Cream Collection”

2026-02-04 AHCラジオ

tomerun

問題案について

原案は何年も前にメモってたものなので元ネタがあったかは覚えておらず

グリッド上を動き回って色々なビット列を作るようなのを考えていた。
短期では特にグリッド問が多くなりがちだし、グラフに変えた

採用理由

- 何かの種類をたくさん作るという設定は、AHCであまりなくちょっと新鮮
- 焼きなましとかビームサーチとかを使わなくとも、工夫次第で
BFSやDFSで貪欲に取っていく方針でけっこう良いスコアが出せる
- LLMに丸投げだとめちゃくちゃ苦戦してた

解答例の紹介

貪欲をベースにして、段階的に改善を重ねることで
ランダム解から赤パフォまでスコアを上げていく例を話していきます

最初の解法：行動1だけ

とりあえず正の得点を取る：この問題ではランダムウォークで一応の点が出る

- 行動1だけ使う。一つ前の頂点以外からランダムに頂点を選んで移動する

33567点 627位

- 自分でスコア計算を実装し、時間いっぱい解答を実行して最もスコアが良かった解を採用する

36687点 618位

以降、「時間いっぱい実行して最も良かった解を採用」は説明から省略します

最初の解法：行動2も使う

行動2を使いたい

いつ・どこで使うかを考えると難しいので、とりあえずランダムにやる

- Wにいるときは1~2%の確率で行動2をする
- 行動2をしないなら、一つ前の頂点以外からランダムに頂点を選んで移動する

86494点 438位

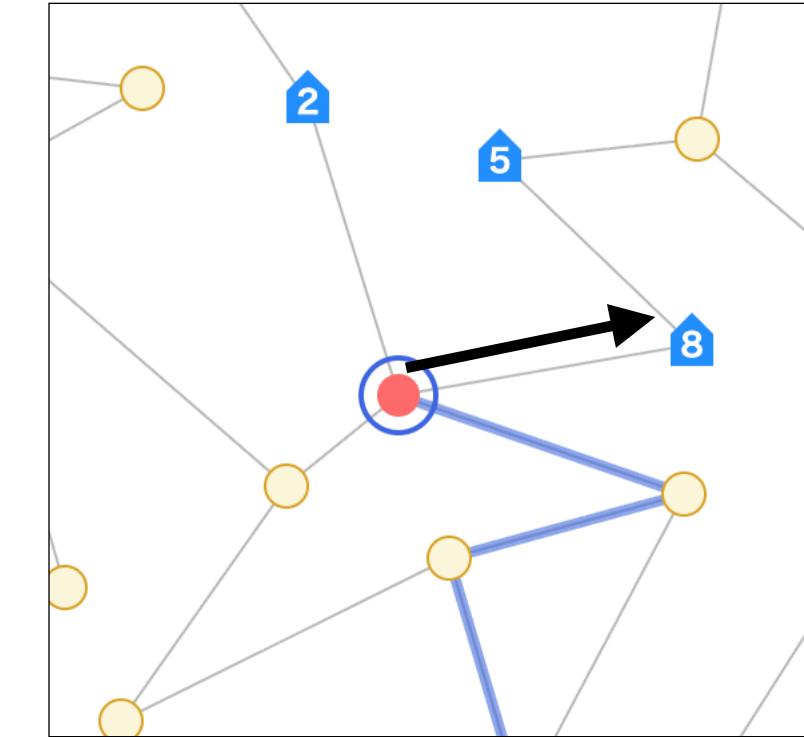
1~2%というのはいろいろ試してみて結果が良かった値

貪欲（1手先）

ランダムができたら次は貪欲をしたい

一番単純そうな貪欲：

- 隣に移動可能な店があって、店に入るとスコアが上がるなら店に移動する
- それ以外なら、
 - Wにいるときは1~2%の確率で行動2をする
 - 行動2をしないなら、一つ前の頂点以外からランダムに頂点を選んで移動する



97260点 380位

貪欲（1手先）

少し工夫：スコアが増えない時に店に入らない

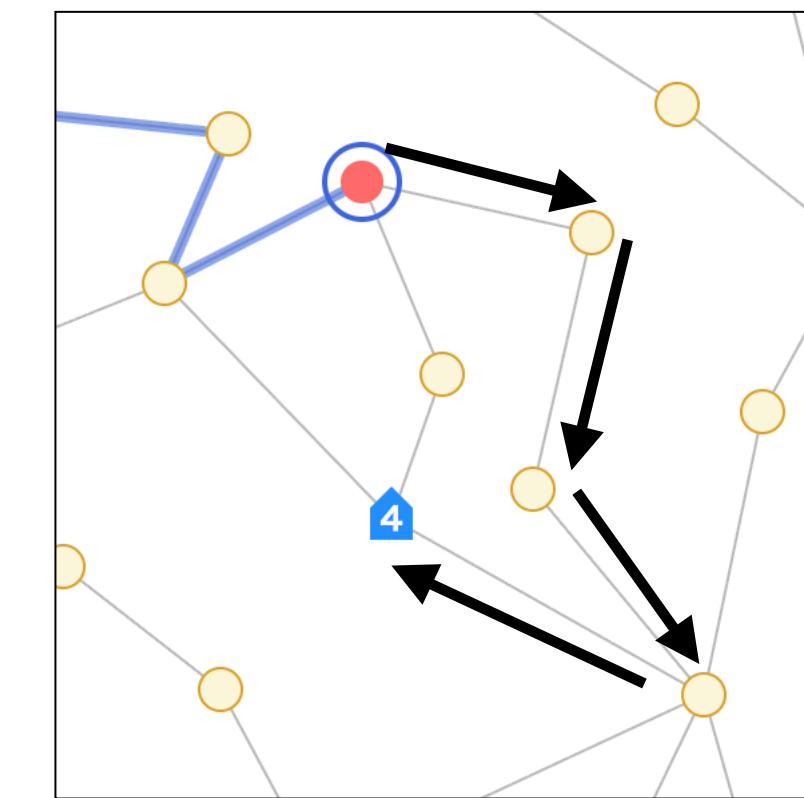
- 隣に移動可能な店があって、店に入るとスコアが上がるなら店に移動する
- それ以外なら、
 - Wにいるときは1~2%の確率で行動2をする
 - 行動2をしないなら、一つ前の頂点以外のアイスの木からランダムに頂点を選んで移動する
 - 移動できるアイスの木がないなら、しかたなく店へ移動する

点が取れないのなら、リセットするよりは長くしといたほうが
新しい文字列になりやすくなる

108463点 335位

貪欲（先読み）

ひとつ隣だけじゃなくもっと先を見る貪欲



- 最大距離8までBFSして、スコアが上がる店までの経路が見つかったらその移動をする
- それ以外なら、
 - Wにいるときは1~2%の確率で行動2をする
 - 行動2をしないなら、一つ前の頂点以外のアイスの木からランダムに頂点を選んで移動する
 - 移動できるアイスの木がないなら、しかたなく店へ移動する

BFSじゃなくてDFSでもあまり変わらないが、短い経路を優先して見つけられる分

BFSのほうが若干良い

反復深化DFSという手法もあるが、グラフの平均分岐数が小さいので効率は悪そう

148998点 157位

貪欲（探索の調整）

赤に変えていくペースは重要

- 最大距離8までBFSして、スコアが上がる店までの経路が見つかったらその移動をする
- それ以外なら、
 - Wにいるときは5~30%の確率で行動2をする
 - 行動2をしないなら、一つ前の頂点以外のアイスの木からランダムに頂点を選んで移動する
 - 移動できるアイスの木がないなら、しかたなく店へ移動する

現状の色の設定のままで頑張って探しすぎるのは良くない

行動2の確率を上げて赤混じりの新しい短い文字列が出てきやすくなるとスコアが上がる

159548点 88位

貪欲（探索の調整）

探索深さの調整 ランダム性を入れるとよくなつた

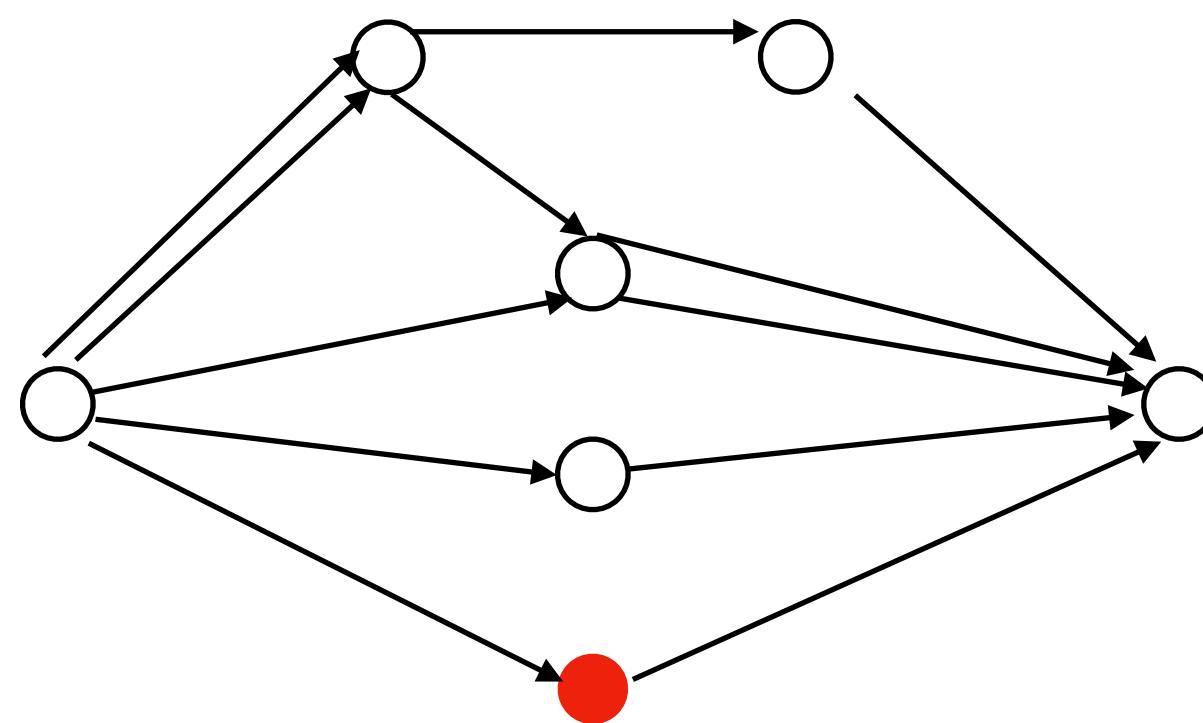
- BFSの最大距離を $\max(\text{rand}(9,11) - \text{現在のコーンの長さ}, 6)$ として、スコアが上がる店までの経路が見つかったらその移動をする
- それ以外なら、
 - Wにいるときは10~50%の確率で行動2をする
 - 行動2をしないなら、一つ前の頂点以外のアイスの木からランダムに頂点を選んで移動する
 - 移動できるアイスの木がないなら、しかたなく店へ移動する

ランダム性によって、現状のまま頑張って文字列を探すのと、色を変えて新しい文字列が出てきやすくなるのと、トレードオフがいい感じになっているんだと思う

164736点 43位

BFSの実装

一般的なBFSでは一度調べた頂点に来たら枝刈りするが、この問題ではそうしてはいけない



これらの経路は全部違う状態
右の頂点に到達した時点でも全部保持すべき

探索の状態は (現在の頂点, 一つ前の頂点, アイスの値) で表され、
これが一致していれば枝刈りできる

ただ、探索済み状態を保持・確認することのオーバーヘッドから、
分岐数が小さいことによって枝刈りせず全探索のほうが速いようでした

AHCでおなじみ 過去改変

- BFSの最大距離を $\max(\text{rand}(9,11) - \text{現在のコーンの長さ}, 6)$ として、スコアが上がる店までの経路が見つかったらその移動をする
- それ以外なら、
 - Wにいるときは10~50%の確率で、現在位置の頂点に前回来た時に行動2をしていたことにする
 - 行動2をしないなら、一つ前の頂点以外のアイスの木からランダムに頂点を選んで移動する
 - 移動できるアイスの木がないなら、したなく店へ移動する

今持っている文字列にも新しい赤が入って新しい文字列を見つけやすくなる

165973点 34位

パス列挙貪欲

1歩ずつ毎回探索するのではなく、店から店への移動を一手として捉える

- 店から店への長さ11以下の経路を全部列挙しておく
- 経路を短い順に見て、スコアが上がる経路があればその移動をする
- なければ、どこか1箇所を赤に変えることでスコアが上がるような最短の経路を探してその移動をする
 - 赤に変える頂点は、前回そこに訪問した時に行動2をしていたこととする
 - それもなければ、どこかの店まで適当に移動する

経路を見る順序を入れ替えつつ時間いっぱい繰り返す
これだけだとスコアは上がっていないが…

158502点 95位

パス列挙貪欲

やっぱり探索の深さにランダム性を入れると良くなった

- 店から店への長さ15以下の経路を全部列挙しておく
- 長さ $\text{rand}(9, 22)$ 以下の経路を短い順に見て、スコアが上がる経路があればその移動をする
- なければ、どこか1箇所を赤に変えることでスコアが上がるような最短の経路を探してその移動をする
 - 赤に変える頂点は、前回そこに訪問した時に行動2をしていたこととする
 - それもなければ、どこかの店まで適当に移動する

赤に変えるのがランダム任せではなく、確実に点を取れるような
変更をしているのが良いのだと思う

169748点 11位

ビームサーチ化

貪欲をビームサーチに発展させるのが定番だが...

- 1つの行動を1遷移にするのは、 $T=10000$ と多くてビーム幅が増やせなさそう
だし、店に入らない行動1や行動2の評価が難しそう
 - (...と思っていたが、2位の方はこの方針だった)
- 店から店に行くまでを1遷移にするのは、スコアが上がる経路を探すのが重い

ということで「次の100ターンの区切りを超えるまでを前ページの貪欲で解く」
を1遷移とする“巨大遷移ビームサーチ”をした

175159点 3位

ビームサーチ

各状態から、調べる経路の順番をランダムに変えつつ時間範囲内で繰り返し貪欲を走らせることで複数の遷移を作る

ビーム幅は100で、各状態から15個くらいの子ができる

状態の評価は、

生スコア + 赤にした数（少ないほうが良い） + ターン数（少ないほうが良い）

多様性確保のため、頂点の色が同じ状態は最高評価の1つだけ残す

本番上位解答

上位は「各頂点をいつ赤に変えるか」を状態に、貪欲で解いたスコアを評価として焼きなます解答が多かった

この方針は差分計算できないしなあ...と思っていたが、イテレーション回数があまり多くなくても高いスコアが出るのは驚き

この問題は色を赤に変えていくタイミングが非常に重要で、それを直接制御するのは強い、ということなのかなと思います

実装tips

アイスの状態は ‘W’ ‘R’ の文字列ではなく、W=0, R=1としたビット列で整数として扱うとよい

そんな長いアイスは作らないので32bit整数で十分

店の在庫の状態の表し方

- アイスの長さごとにsetに入れる
- ビット列の先頭に1を追加することで長さを区別し、一つのsetに入れる
- 上と同様に長さを区別し、サイズ $2^{(最大アイス長+1)}$ のbool配列で管理する

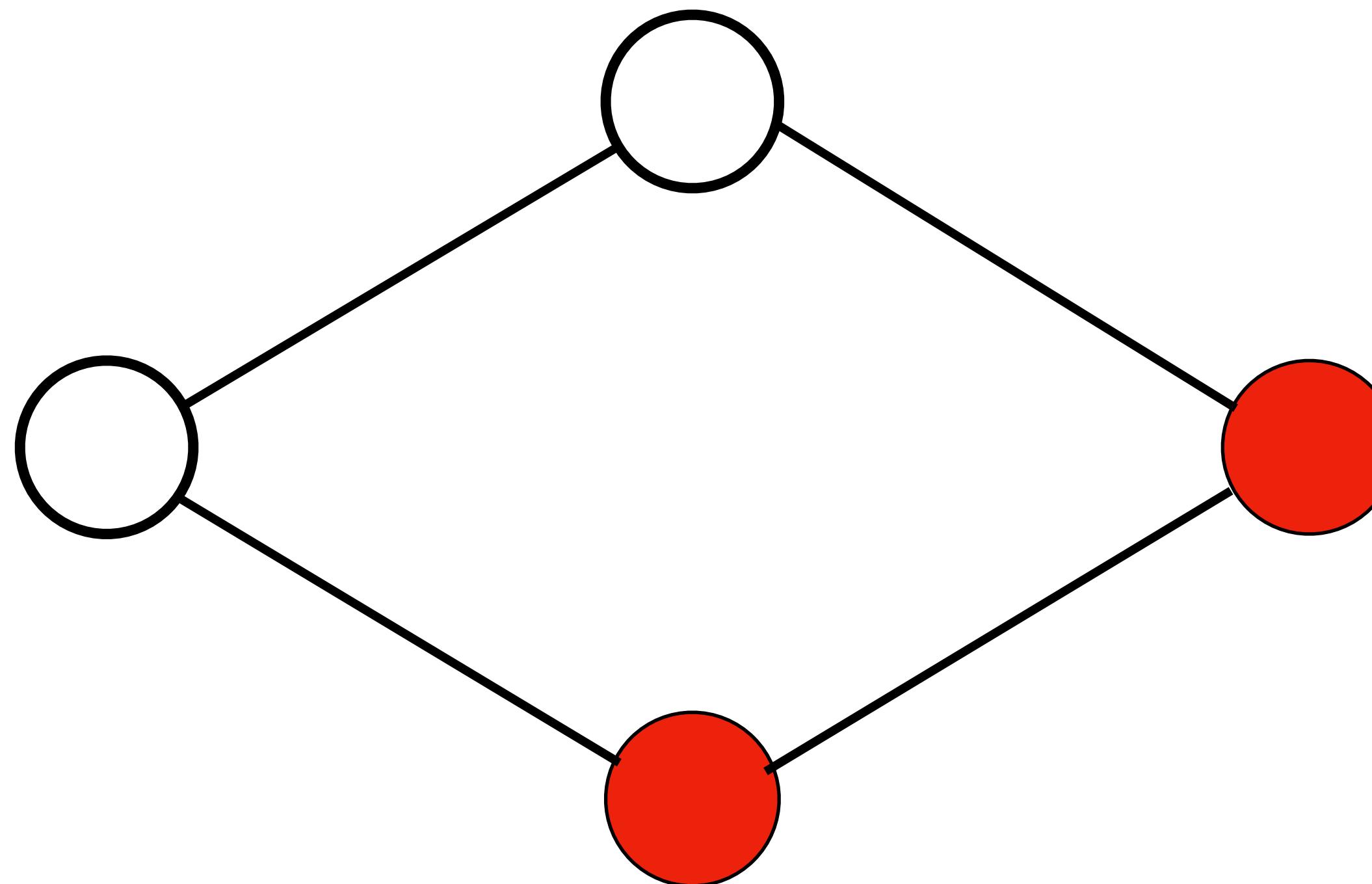
(empty)	1
W	10
R	11
WW	100
RW	110
WR	101

ビームサーチ系なら2番目、焼きなまし系なら3番目が良さそう

余談

Uターン禁止制約

Uターン禁止がないと、長さ4のサイクルで 白-白-赤-赤 にしたら
任意の文字列を簡単に作れてしまう



仕様の拡張

「店の頂点を、いくつ・どこにするかを自由に決められる」

という仕様も面白そう

8時間コンテストだったらちょうど良いくらいか

店の数は何個くらいが一番良いのかはあまり直感的にわからない

(少なくとも10よりは多いはず)

「例を見る」の出力

行動1だけのランダムウォークより弱くなるよう、
それでいて自明すぎずそれなりの点を取るような解答を作った

$g[v]$ を、 v に隣接する頂点のリストとする。
現在位置を x 、 $g[x]$ の中で直前の頂点以外で先頭にある頂点を y とする。
 y を $g[x]$ の末尾へ移し、 y へ向かう 行動1 をする
(これに加えて、序盤では適当に何回か行動2をする)

これで、長い周期でグラフを巡回するような動きになる