

最短路の基本

1 導入

この講義で最重要なのは、最短路では「1本進む重さがみな同じか」「辺ごとに重みが違うか」を最初に見分けることです。

最短といっても、辺の本数を最小にしたいのか、重みの和を最小にしたいのかで使う道具が変わります。こ

こを曖昧にすると、BFS でよい問題とダイクストラ法を使うべき問題を混同します。

2 用語と定義

最短路とは、ある始点から終点までの道のうち、長さが最小のものです。

重みとは、辺についた距離や費用のような値です。

3 方針

まず「重みなし」なら BFS で距離を層として広げます。「重みあり」で負でないなら、現時点で最小の距離をもつ頂点から確定していきます。

4 直感的な説明

重みがないなら、BFS の波が1歩ずつ広がるので、最初に着いた時点でその歩数が最短です。いっぽう、重みがあると「1本進む」だけでは距離を比べられないので、いちばん短く着ける候補から順に確定していく必要があります。

5 厳密な説明

5.1 1. 重みなしの場合

ここでは BFS の層構造をそのまま使うので、キューでどう探索が広がるかがまだ曖昧なら、先にこちらを見ないと理解しやすくなります。

→ [講義](#) [DFS と BFS の基本](#) [lecture](#) [information](#) [algorithm](#)
<https://study.bem130.com/lecture/information/algorithm/search/DFS と BFS の基本-講義/>

BFS では、始点から距離1の頂点、つぎに距離2の頂点という順で探索します。したがって、最初に到達したときの距離が最短です。

5.2 2. 重みありの場合

辺の重みが負でないなら、ダイクストラ法で未確定の頂点のうち距離が最小のものを選んで確定していきます。

5.3 3. 使い分け

重みなしなら BFS、重みありで負でないならダイクストラ法、という整理が基本です。

6 見分け方

- 辺の本数を最小にする問題なら BFS を疑います。
- 時間、費用、距離のような重みが辺についているなら、BFS だけでは足りません。
- 「最初に着いたら最短」が言えるのは、BFS の層構造が使えるときです。

7 最終形

重みなし最短路 ⇒ BFS

重みあり最短路 ⇒ ダイクストラ法の候補

8 一言でいうと

- 最短路では、まず「何を長さとしみなすか」を決め、それに応じて BFS とダイクストラ法を使い分けます。