

# ダイクストラ法・BFS・A\* を図で 理解する課題（時間補習課題）

## 【学習目標】

この課題では次のことができるようになります。

- BFS（幅優先探索）の基本と動き方を説明できる
- ダイクストラ法の仕組みを理解できる
- A\* が「何を追加したアルゴリズム」か説明できる
- 図を見て3つの探索アルゴリズムを比較できる
- ゲーム開発での使い分けが説明できる

## ☒ 【課題図（グラフ構造）】

```
(1)   (1)   (1)
[S] ---- [A] ---- [B] ---- [G]
  \
   \ (5)
    \
     [C]
      \
       \ (1)
        \
         [B] (同じB)
```

- S：スタート
- G：ゴール
- 数字：その道を通るコスト

## ◆ ① BFS（幅優先探索）とは？ （穴埋め）

BFS は、  
スタートから（ ）の順に探索していくアルゴリズムで、  
道のコストが（ ）ときに最短経路を見つけることができる。

選択肢：

- 近いノード
- 深さ
- 同じ
- バラバラ

## ◆ ② ダイクストラ法とは？（穴埋め）

ダイクストラ法は、スタートからの（ ）を更新しながら、（ ）が最も小さいノードを順に確定する方法である。

選択肢：

- 最短距離
- コスト
- 深さ
- 未確定ノード

## ◆ ③ A\* アルゴリズムとは？（穴埋め）

A\* は、ダイクストラ法に（ ）を追加した方法で、より（ ）ゴールに到達することを目的としている。

選択肢：

- ヒューリスティック
- ランダム
- 早く
- 正確に

## ◆ ④ 図を見て考える：BFS / ダイクストラ / A\* の動き

Q1. BFS はどのルートを最初に見つける？

BFS が最初に見つけるルートは（ ）  
理由： \_\_\_\_\_

Q2. ダイクストラ法が求める“最短経路”は？

- 経路：（ ）
- 合計コスト：（ ）

ヒント：

- $S \rightarrow A \rightarrow B \rightarrow G = 1 + 1 + 1 = 3$
- $S \rightarrow C \rightarrow B \rightarrow G = 5 + 1 + 1 = 7$

Q3. A\* の式を完成させよう

A\* の評価値  $f(n)$  は次の式で求める：

$$f(n) = g(n) + h(n)$$

- $g(n)$  : Sからそのノードまでの実コスト
- $h(n)$  : ゴールまでの ( )

選択肢：

- 予想距離 (予想値)
- 完全一致する距離
- 乱数

## ◆ ⑤ BFSの理解を深める

### 追加Q1：BFSはどんな順番で探索を進める？

AとBどちらかを選んで、理由を書くこと。

- A：スタート地点から「近いノード」から順番に調べる
- B：コストが小さい道を優先して調べる

回答：A/B (理由：\_\_\_\_\_)

### 追加Q2：図を使ってBFSの探索順を書こう

例：S → A → B → C → Gのように記述。

探索順：\_\_\_\_\_

### 追加Q3：なぜBFSは「コスト同じのときだけ」最短経路を求められるのか？

穴埋め：

BFSは、道のコストがすべて ( ) とき、スタートからの ( ) の順に探索するため、最短経路が求まる。

選択肢：

- 同じ
- バラバラ
- 深さ (段階)
- 重さ

### 追加Q4：コストがバラバラのときにBFSが失敗する例

BFSはどちらを先に探索する？

- ルート1：S → A → B → G (合計3)
- ルート2：S → C → B → G (合計7)

回答：\_\_\_\_\_  
理由：\_\_\_\_\_

### 追加Q5：ゲーム開発でBFSが役立つ場面は？

番号を選び、理由を書く。

1. 敵から一定距離以内に逃げたいとき
2. コストがバラバラのマップで探索したい

3. 同じコストのタイルで広いマップを探索するとき

選んだ番号： \_\_\_\_

理由： \_\_\_\_\_

## 追加Q6：BFSの弱点

図のように“遠回りだが軽い道”があるとき、BFSはどんな失敗をするか？

回答： \_\_\_\_\_

## 追加Q7：BFSを1行でまとめよ

「BFSは \_\_\_\_\_ を優先し、コストが \_\_\_\_\_ ときに最短経路を求められる」

## 追加Q8：BFSとダイクストラの共通点・相違点（穴埋め）

- 共通点：どちらも（ ）に沿って探索し、最終的に（ ）を求める
- 違い：BFSは（ ）が同じときに最短になるが、ダイクストラ法は（ ）が0以上であれば最短を求められる

# ◆ ⑥ 3つのアルゴリズムの違いを1行ずつまとめよう

- **BFS**：道のコストが（ ）とき、 \_\_\_\_\_ の順に探索する
- **ダイクストラ法**：コストが0以上なら \_\_\_\_\_ を必ず見つけられる
- **A\***：ヒューリスティックを使って \_\_\_\_\_ 探索する

# ◆ ⑦ ゲーム開発での使い分け

1. 草原でコストが全部同じ  
→（ ）  
理由： \_\_\_\_\_
2. コストがバラバラの道（泥・坂・砂）  
→（ ）  
理由： \_\_\_\_\_
3. ゴール（敵の位置）が分かっている  
→（ ）  
理由： \_\_\_\_\_

# ◆ ⑧ 最終まとめ（作文）

ダイクストラ法・BFS・A\*の違いを、初心者にもわかるよう詳細に説明せよ。

