

深さ優先探索のトレース

表1 BFS と DFS

特徴	幅優先探索 (BFS)	深さ優先探索 (DFS)
探索のイメージ	近くから順に、層を広げるように探索	行けるところまで深く、行き止まりまで進む
得意なこと	最短経路の発見 (辺の重みが全て同じ場合)	連結性の確認, 全てのノードの訪問
使用するデータ構造	キュー (Queue): 先入れ先出し	スタック (Stack) または 再帰関数: 後入れ先出し
メモリ消費	探索範囲が広がると大きくなる傾向がある	比較的少ないメモリで済むことが多い

DFS は以下のような手順で動作します。

(スタックを用いる場合)

1. **準備:** 探索開始ノード s をスタック (stack) に push (追加) し, 訪問済みとしてマークする。
2. **ループ:** スタックが空になるまで, 以下の手順 3~5 を繰り返す。
3. **取り出し:** スタックのトップ (一番上) からノード v を pop (取り出し) する。
4. **処理:** ノード v を訪問 (探索) する。
5. **探索:** ノードに v 隣接するまだ訪問していないすべてのノードを, スタックにプッシュ (追加) し, それぞれを訪問済みとしてマークする。前進できる未マークの頂点がない場合, 一つ前の頂点に戻ります (バックトラック)。
6. 開始地点 s まで戻り, それ以上後退できなくなった時点で終了します。

(再帰を用いる場合)

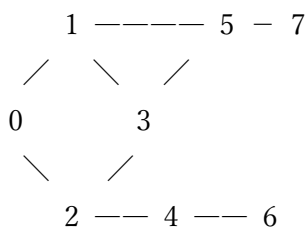
1. **開始ノードを訪問済み**にする。
2. 現在ノードに隣接する未訪問ノードを一つ選ぶ。
3. 選んだノードへ移動し, **再帰的に手順 1~3 を実行**する (ステップイン)。
4. 未訪問ノードがなくなったら, **一つ前に戻る** (バックトラック/ステップアウト)。

表 2 各頂点の状態の表示

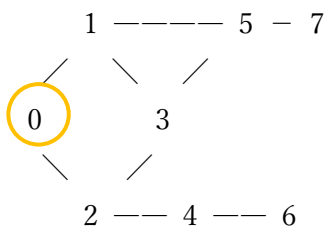
色	状態
赤	その頂点がすでに 訪問済み であることを表す（ネットサーフィンで言えば、ページを読了済みであること）
橙	その頂点がすでに発見されていて、これから 訪問予定 であることを表す
白	その頂点が 未発見 であることを表す。

(2) DFS の手順

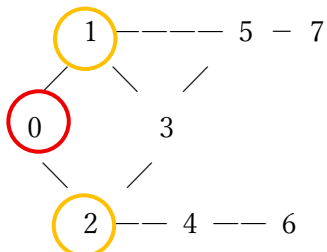
(手順 0) 例えば、下の図に対して DFS の動きを確認します。すべての頂点は**「白（未発見）」**です。スタックは空です。



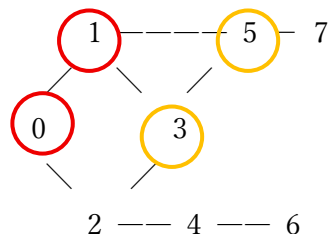
(手順 1) 頂点 **0** をスタックに入れます。頂点 **0** の状態を「橙（訪問予定）」にします。



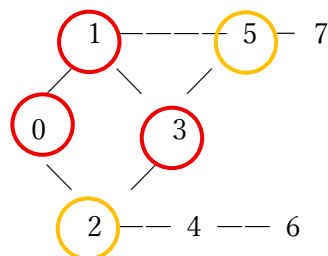
(手順 2) スタックから最新の要素 **0** を取り出します。頂点 **0** を「赤（訪問済み）」にします。頂点 **0** に隣接する白い頂点 **1, 2** をスタックに入れます。頂点 **1, 2** を「橙（訪問予定）」にします。



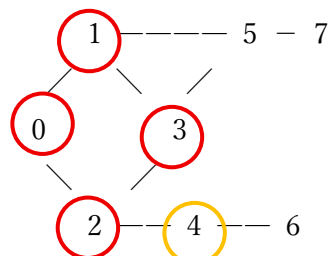
(手順3) スタックの先頭から 1 を取り出します (0 から見て一番新しい発見)。頂点 1 を「赤 (訪問済み)」にします。頂点 1 に隣接する白い頂点 3, 5 をスタックに入れます。頂点 3, 5 を「橙 (訪問予定)」にします。



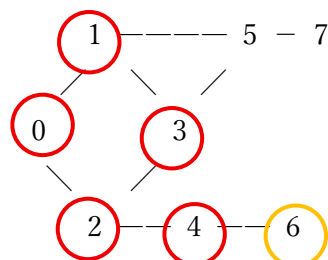
(手順4) スタックの先頭から 3 を取り出します (1 から見て一番新しい発見)。頂点 3 を「赤 (訪問済み)」にします。頂点 3 に隣接する白い頂点 2, 5 をスタックに入れます。頂点 2, 5 を「橙 (訪問予定)」にします。



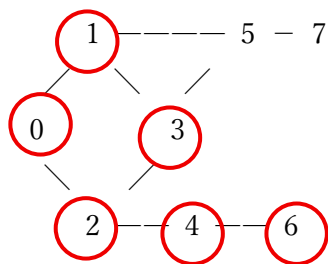
(手順5) スタックの先頭から 2 を取り出します (3 から見て一番新しい発見)。頂点 2 を「赤 (訪問済み)」にします。頂点 2 に隣接する白い頂点 4 をスタックに入れます。頂点 4 を「橙 (訪問予定)」にします。



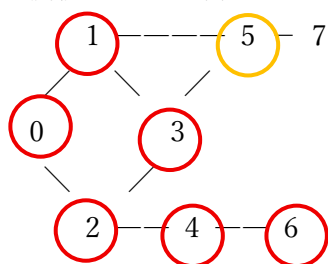
(手順6) スタックの先頭から 4 を取り出します (2 から見て一番新しい発見)。頂点 4 を「赤 (訪問済み)」にします。頂点 4 に隣接する白い頂点 6 をスタックに入れます。頂点 6 を「橙 (訪問予定)」にします。



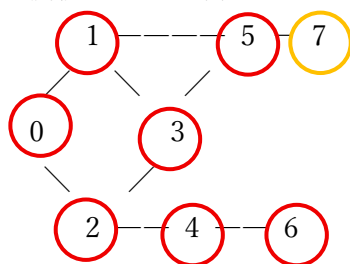
(手順 7) 6 を「赤 (訪問済み)」にします。6 から「橙 (訪問予定)」が見つかるまで戻る。



(手順 10) スタックの先頭から 3 を取り出します。頂点 3 を「赤 (訪問済み)」にします。頂点 3 に隣接する白い頂点 5 をスタックに入れます。頂点 5 を「橙 (訪問予定)」にします。



(手順 11) スタックの先頭から 5 を取り出します。頂点 5 を「赤 (訪問済み)」にします。頂点 5 に隣接する白い頂点 7 をスタックに入れます。頂点 7 を「橙 (訪問予定)」にします。



(手順 12) スタックを空にします。

(手順 13) 頂点 7 を「赤 (訪問済み)」にします。

表3 DFS のトレース

手順	取り出した頂点	訪問済 (赤)	操作	スタックの中身 (右が TOP)	説明
1	-	-	開始	[0]	頂点 0 からスタート
2	0	{0}	前進	[0, 1]	0 に隣接する 1 へ (0→1)
3	1	{0, 1}	前進	[0, 1, 3]	1 に隣接する 3 へ (1→3)
4	3	{0, 1, 3}	前進	[0, 1, 3, 2]	3 に隣接する 2 へ (3→2)
5	2	{0, 1, 3, 2}	前進	[0, 1, 3, 2, 4]	2 に隣接する 4 へ (2→4)
6	4	{0, 1, 3, 2, 4}	前進	[0, 1, 3, 2, 4, 6]	4 に隣接する 6 へ (4→6)
7	6	{0, 1, 3, 2, 4, 6}	後退	[0, 1, 3, 2, 4]	6 は行き止まり。6 を Pop し て 4 に戻る
8	4	{0, 1, 3, 2, 4, 6}	後退	[0, 1, 3, 2]	4 も未訪問なし。4 を Pop し て 2 に戻る
9	2	{0, 1, 3, 2, 4, 6}	後退	[0, 1, 3]	2 も未訪問なし。2 を Pop し て 3 に戻る
10	3	{0, 1, 3, 2, 4, 6}	前進	[0, 1, 3, 5]	3 に隣接する未訪問の 5 へ (3→5)
11	5	{0, 1, 3, 2, 4, 6, 5}	前進	[0, 1, 3, 5, 7]	5 に隣接する 7 へ (5→7)
12	7	{0, 1, 3, 2, 4, 6, 5, 7}	探索完了	[0, 1, 3, 5]	
13	なし	{0, 1, 3, 2, 4, 6, 5, 7}	全スタック pop	[]	