データ構造とアルゴリズムII

第7回
幅優先／深さ優先探索／トポロジカルソート

22. 基本的グラフアルゴリズム

無向グラフ

・5個の頂点と7本の辺からなる無向グラフ

無向グラフG=(V,E), Vは頂点集合, Eは辺集合. Eの要素は頂点のペア\{u,v\}によって表される. \{u,v\}と\{v,u\}は同じ辺を意味する.

隣接リスト

・各頂点に関して, 隣接する(直接, 込で結ばれた)頂点集合をリストで表現

隣接行列


有向グラフ

・5個の頂点と8本の込からなる有向グラフ

有向グラフG=(V,E), Vは頂点集合, Eは込集合. Eの要素は頂点の順序付きペア(u,v)によって表される. (u,v)と(v,u)は異なる. (u,u)なる込(セルフループ)を許す.
隣接リスト
• 各頂点に関して、隣接する（自身からその頂点に向かう有向辺がある）頂点集合をリストで表現

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

隣接行列
• 隣接関係を$|V| \times |V|$の行列で表現。$A[u][v]=1/0$により辺$(u, v)$の存在/非存在を示す。
対称とは限らない。

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

幅優先探索（22.2）
• 幅優先探索（Breadth-first search, BFS）は最も単純なグラフ探索アルゴリズムの一つ。
• 始点$s$が与えられたら、$s$から到達可能な節点を系統的に探索する。
• BFSはqueueを用いて実現可能。
• Primの最小全域木アルゴリズムやDijkstraの単一始点最短路アルゴリズムはBFSの一種と考えられる。

BFSによる探索
• まだ訪れていない節点の色は白。
• 初めて発見された節点は灰色（図中では緑）。
• 近傍がすべて発見されたら黒（図中では赤）。
• 灰色になる時に、始点$s$からの距離と、親となる節点を決定。
• 有向でも無向でも動く（以下は無向連結グラフで説明）。

BFSによる探索例

(a) 1 2 3 4 5
(b) 1 2 3 4 5
(c) 1 2 3 4 5
BFSによる探索例

1. for each vertex u ∈ G.V-{s}
2. u.color = WHITE
3. u.d = ∞
4. u.π = NIL
5. s.color = GRAY
6. s.d = 0
7. s.π = NIL
8. Q = empty
9. Enqueue(Q,s)
10. while Q≠empty
11. u = Dequeue(Q)
12. for each v ∈ G.Adj[u]
13. if v.color == WHITE
14. v.color = GRAY
15. v.d = u.d + 1
16. v.π = u
17. Enqueue(Q,v)
18. u.color = BLACK

Time Complexity: O(|E|+|V|)

BFSの性質
・幅優先探索木が構成され、始点からの最短経路が与えられる。幅優先探索木は複数存在し得るが、始点からの最短経路は同じ。

有向／非連結グラフのBFS
・基本的な動作は同様
・queueが空になっても、まだ未発見の頂点が残っているれば、その一つを選んでqueueに加える

演習：幅優先探索
・上のグラフでの幅優先探索木を示せ。

演習：幅優先探索
・上のグラフでの幅優先探索木を示せ。

深さ優先探索(22.3)

- 可能ならばグラフの“より深い部分”を探索する。
- 未探索の外向辺がある頂点中で、最後に発見した頂点vからの未探索の外向辺を探索。
- vの辺をすべて探索し終わると、vを発見した時に通った辺を“バックラック(逆戻り)”し、vの直前の頂点の未探索の外向辺を探索。
- この処理を元の始点から到達可能なすべての頂点を発見するまで続ける。
- 未発見の頂点が残ったら、そのうち1つを新たな始点として選択し、そこから探索を続ける。
- 有向でも無向でも動く(以下は有向グラフで説明)

深さ優先探索アルゴリズム

- 深さ優先探索アルゴリズムは各頂点に時刻印をつける。
- どの頂点vも2つの時刻印を持つ。
- 最初の時刻印v.dはvを最初に発見したときにつける。このとき、頂点は灰色に塗られる。
- 2番目の時刻印v.fは、vの隣接リストを調べ終えたときに記録する。このとき、頂点は黒色に塗られる。
- 時刻印をつけたら、時計の針が進む。
- 時刻印をつけた頂点は、その頂点の色が時刻印の色と同じである。
- またすべての頂点について次式が成り立つ。
- 顶点vの色は時刻印vo以前は白、頂点vの色は灰色、それ以後は黒である。

DFSの適用例

(a) 发見時刻
(b) DFSの適用例
(c) DFSの適用例
(d) DFSの適用例
(e) DFSの適用例
(f) DFSの適用例
DFSアルゴリズム

DFS(G)
1. for each vertex u ∈ G.V
2. do u.color = WHITE
3. u.π = NIL
4. time = 0
5. for each vertex u ∈ G.V
6. if u.color == WHITE
7. DFS-Visit(G,u)

DFS-Visit

DFS-Visit(G, u)
1. time = time+1 //u has just been discovered
2. u.d = time
3. u.color = GRAY
4. for each v ∈ G.Adj[u]
5. if v.color == WHITE
6. v.π = u
7. DFS-Visit(G, v)
8. u.color = BLACK //DFS-Visit(G, u) is done
9. time = time + 1
10. u.f = time

深さ優先探索の性質

・DFSの実行時間はO(|V|+|E|).
・“木辺”によって深さ優先森を形成. 辺の選択順序が決まれば, これはユニーケに決まる.

DFS: 辺の種類

・DFSによって, 辺がいくつかの種類に分類される:
  - 木辺: 深さ優先森中の木の辺, 新しい(白)頂点に至る辺
  - 後退辺: 木辺以外で, 祖先に向かう辺, 灰色頂点から灰色頂点に至る辺
  - 前進辺: 木辺以外で, 子孫に向かう辺, 灰色から黒に至る辺
  - 横断辺: その他, 灰色から黒に至る辺.
・多くのアルゴリズムでは木辺と後退辺が重要

DFSの例
演習: 深さ優先探索
・上のグラフでの深さ優先探索を行った場合、各枝の種類（木辺、後退辺、前進辺、横断辺）を示せ。ただし、頂点の選択順／隣接リストはアルファベット順とする。

演習: DFS: 辺の種類
・Gが無向グラフの時、DFSは木辺か後退辺のみを生成することを示せ。

証明:
辺(u, v)を考える
一般性を失うことなしにu.d < v.dを仮定。
v.f<u.fのはずです。
uからvに向けて、この辺を発見した
なら、この時点vは白であるはずです
(vが既発見なら、vからuに向けて、すぐにこの辺を発見していないといけない)
この場合、辺は木辺
vからuに向けて、この辺を発見したら、
この時点vは灰色、uも灰色、よって辺は後退辺
トポロジカルソート

- 閉路のない有向グラフ(Directed Acyclic Graph, DAG)\(G = (V,E)\)のトポロジカルソートとは、\(G\)が辺\((u,v)\)を含んでいれば\(u\)が\(v\)より先に現れるような、全頂点への線形順序を求めること。
- 深さ優先探索を用いるとDAGのトポロジカルソートは簡単に得られる。
- グラフのトポロジカルソートは、水平な直線上に頂点を並べて、どの有向辺も左から右へ向かうように全頂点を並べることに対応。

トポロジカルソートのアルゴリズム

\[
\text{TOPOLOGICAL-SORT}(G) \\
\text{1} \quad \text{DFS}(G) \text{を呼び出して、各頂点} v \text{の終了時刻} v.f \text{を計算} \\
\text{2} \quad \text{終了した頂点を、連結リストの先頭に挿入} \\
\text{3} \quad \text{return 頂点の連結リスト}
\]

深さ優先探索が\(O(V+E)\)時間でき、\(|V|\)個の頂点を連結リストの先頭に挿入するのは各々\(O(1)\)時間でできるから、トポロジカルソートは\(O(V+E)\)時間で実行できる。

演習:トポロジカルソート:正しさの証明

以下を証明せよ:
有向グラフ\(G\)が閉路を持たないのは、\(G\)を深さ優先探索したときに後退辺を生成しないときであり、かつそのときに限る。

\[
\begin{align*}
&\Rightarrow \text{後退辺}(u,v) \text{が生じると仮定すると、深さ優先探索森において、}u \text{は}v \text{の先祖となるので、}v \text{から}u \text{への経路があることになるが、これに後退辺}(u,v) \text{を加えると閉路になる。} \\
\end{align*}
\]

\[
\begin{align*}
&\Leftrightarrow \text{Gが閉路を含むと仮定する。}v \text{を}u \text{において最初に発見される頂点とし、}(u,v) \text{において入る辺とする。時刻} v.d \text{では}v \text{から}u \text{に至る白頂点の経路が存在する。よって、}u \text{は深さ優先探索森においての子孫になる。したがって、}(u,v) \text{は後退辺である。}
\end{align*}
\]

演習:トポロジカルソート:正しさの証明

\[
\begin{align*}
&\text{以下を証明せよ(補題22.11):} \\
&\text{有向グラフGが閉路を持たないのは、Gを深さ優先探索したときに後退辺を生成しないときであり、かつそのときに限る。} \\
&(\text{証明}) \\
&\Rightarrow \text{後退辺}(u,v) \text{が生じると仮定すると、深さ優先探索森において、}v \text{は}u \text{の先祖となるので、}G\text{に}v \text{から}u \text{への経路があることになるが、これに後退辺}(u,v) \text{を加えると閉路になる。} \\
&\Leftrightarrow \text{Gが閉路を含むと仮定する。}v \text{を}u \text{において最初に発見される頂点とし、}(u,v) \text{において入る辺とする。時刻} v.d \text{では}v \text{から}u \text{に至る白頂点の経路が存在する。よって、}u \text{は深さ優先探索森においての子孫になる。したがって、}(u,v) \text{は後退辺である。}
\end{align*}
\]

演習:トポロジカルソート:正しさの証明

定理22.12

\[
\text{TOPOLOGICAL-SORT}(G) \text{は、閉路のない有向グラフGをトポロジカルソートする。}
\]

(証明)

\[
\begin{align*}
&\text{与えられた有向グラフGに関してDFSを実行して各頂点の終了時刻を計算する。任意の異なる頂点}u,v \text{に対して、}G\text{の途中で}v \text{から}u \text{への辺があるとき、}v \text{が白のとき、}u \text{の子孫になるので、}v \text{が白のとき仮に}v < u.f \text{としたがって閉路のない有向グラフの任意の辺に対して}v.f < u.f \text{が成り立つので証明が完成。}
\end{align*}
\]

トポロジカルソート

- 閉路のない有向グラフ(Directed Acyclic Graph, DAG)\(G = (V,E)\)のトポロジカルソートとは、\(G\)が辺\((u,v)\)を含んでいれば\(u\)が\(v\)より先に現れるような、全頂点への線形順序を求めること。
- 深さ優先探索を用いるとDAGのトポロジカルソートは簡単に得られる。
- グラフのトポロジカルソートは、水平な直線上に頂点を並べて、どの有向辺も左から右へ向かうように全頂点を並べることに対応。

トポロジカルソートのアルゴリズム

\[
\text{TOPOLOGICAL-SORT}(G) \\
\text{1} \quad \text{DFS}(G) \text{を呼び出して、各頂点} v \text{の終了時刻} v.f \text{を計算} \\
\text{2} \quad \text{終了した頂点を、連結リストの先頭に挿入} \\
\text{3} \quad \text{return 頂点の連結リスト}
\]

深さ優先探索が\(O(V+E)\)時間でき、\(|V|\)個の頂点を連結リストの先頭に挿入するのは各々\(O(1)\)時間でできるから、トポロジカルソートは\(O(V+E)\)時間で実行できる。
演習: トポロジカルソート
以下のグラフのトポロジカルソートを求めよ。