「グラフと組み合わせ」期末試験問題

–2016 年度前期–

2016/8/4

解答例

1　数学的帰納法

以下の自然数 n に対する恒等式を、数学的帰納法を用いて証明しなさい。

$$\sum_{k=1}^{n}(2k-1)^2=\frac{1}{3}n\left(4n^2-1\right) \tag{1.1}$$

解答例

1. $n=1$ の場合、左辺は

   $$(2-1)^2=1^2=1$$

   右辺は

   $$\frac{1}{3}\times 1\times(4-1)=1$$

   となり、式 (1.1) が成り立つ。

2. ある n について式 (1.1) が成り立つと仮定し、$n+1$ の場合を調べる。

   $$\begin{aligned}\sum_{k=1}^{n+1}(2k-1)^2 &= \sum_{k=1}^{n}(2k-1)^2+(2(n+1)-1)^2\\ &= \frac{1}{3}n\left(4n^2-1\right)+(2n+1)^2\\ &= \frac{1}{3}(2n+1)\left[n(2n-1)+3(2n+1)\right]\\ &= \frac{1}{3}(2n+1)(n+1)(2n+3)\\ &= \frac{1}{3}(n+1)(2(n+1)-1)(2(n+1)+1)\\ &= \frac{1}{3}(n+1)\left(4(n+1)^2-1\right)\end{aligned}$$

   つまり式 (1.1) が $n+1$ の場合にも正しいことを導出することができた。以上、数学的帰納法により式 (1.1) が正しいことを証明した。

2　深さ優先探索

有向グラフ $G=(V,A)$ に対する深さ優先探索を考える。探索を始点を v_0、既に探索した頂点の集合を L とし、深さ優先探索は以下のような再帰アルゴリズム 2.1 を search($v_0,\{v_0\}$) として呼ぶことで実行できる。

```
search(v, L){
    forall ( a ∈ δ⁺v ) { //v を始点とする全ての弧
        w = ∂⁻a //弧 a の終点
        if ( w ∉ L ) {
            L ← L ∪ {w} //頂点 w を L に追加
            search (w, L)
        }
    }
}
```

Algorithm 2.1: 深さ優先探索。//はコメント。

2.1 深さ優先探索の実行

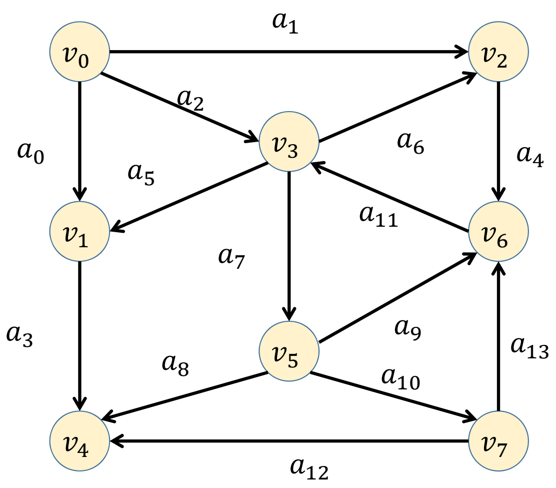


図 1: Algorithm 2.1 を用いる対象グラフ。

図 1 のグラフに対して、v_0 を始点として、深さ優先探索を実行しよう。最初に search$(v_0, \{v_0\})$ を呼び出す。これを $[v_0, \{v_0\}]$ と表記する。$[v_0, \{v_0\}]$ の中から $[v_1, \{v_0, v_1\}]$ が呼び出される。$[v_1, \{v_0, v_1\}]$ の処理が終わると、$[v_2, L]$ が呼び出されるであろう (ここで、二番目の引数は $[v_1, \{v_0, v_1\}]$ の処理後に定まるために、L と表記している)。$[v_3, L]$ が $[v_0, \{v_0\}]$ の中からの中から呼ばれるかは、ここでは不明としておく。

このような呼び出し順序を図 2 にまとめることにする。この図を完成させなさい (図中の L は、適切な頂点の集合に置き換えること)。なお、複数の弧の選択肢がある場合には、弧の番号の小さいほうを優先しなさい。

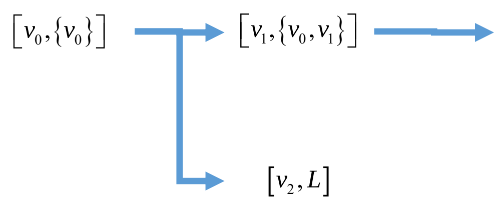
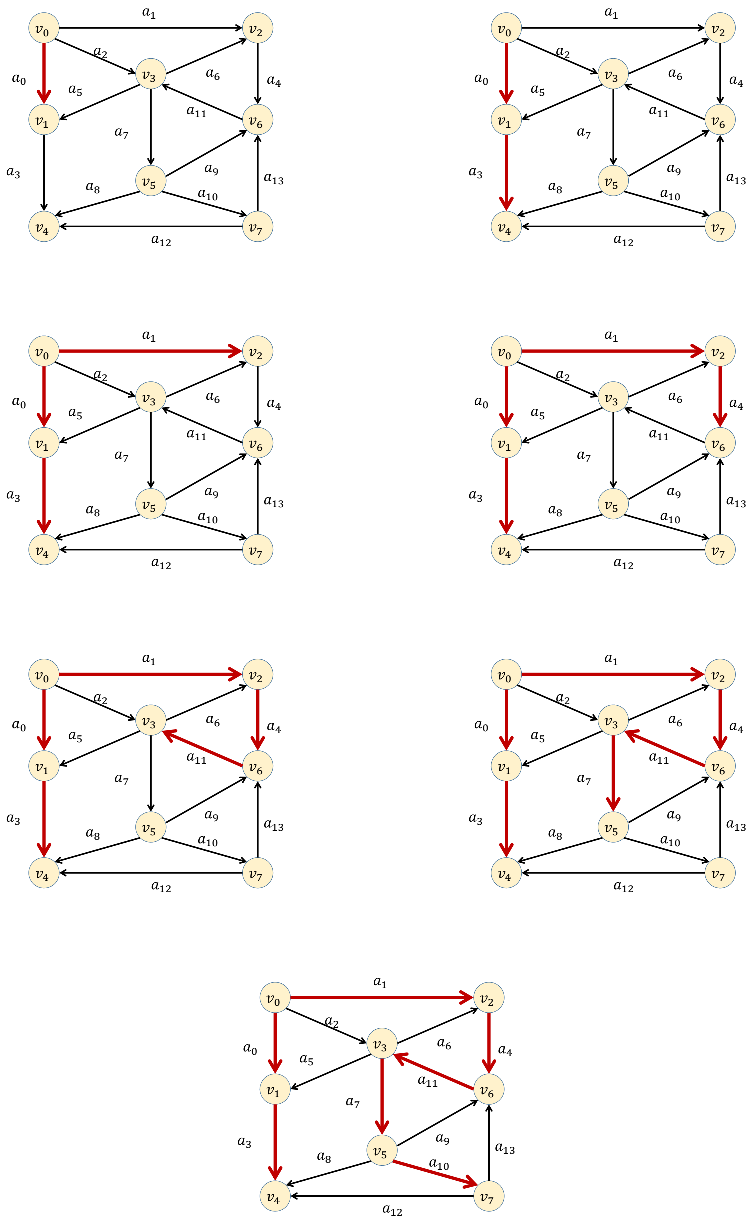


図 2: グラフ 1 を対象に深さ優先探索を実施した際の、アルゴリズムの再帰呼び出し状況。

解答例グラフ１を対象に深さ優先探索を実施した際の、アルゴリズムの再帰呼び出し状況

$$
\begin{tikzcd}
{\left[v_0,\{v_0\}\right]} \arrow[r] \arrow[dr] & {\left[v_1,\{v_0,v_1\}\right]} \arrow[r] & {\left[v_4,\{v_0,v_1,v_4\}\right]} \\
& {\left[v_2,\{v_0,v_1,v_2,v_4\}\right]} \arrow[r] & {\left[v_6,\{v_0,v_1,v_2,v_4,v_6\}\right]} \arrow[d] \\
& & {\left[v_3,\{v_0,v_1,v_2,v_3,v_4,v_6\}\right]} \arrow[dll] \\
{\left[v_5,\{v_0,v_1,v_2,v_3,v_4,v_5,v_6\}\right]} \arrow[r] & {\left[v_7,\{v_0,v_1,v_2,v_3,v_4,v_5,v_6,v_7\}\right]} &
\end{tikzcd}
$$

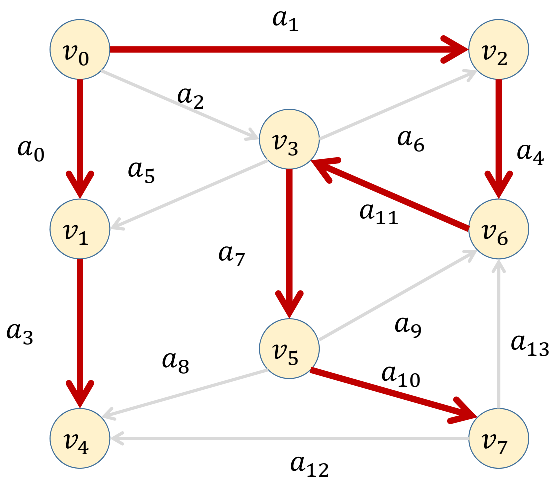


v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13}

2.2　探索木

深さ優先探索で得られた探索木を示しなさい。

解答例



3　最小木

無向グラフ $G = (V, A)$ の各弧 $a \in A$ に対して、正の実数によって重み $(w : A \to R)$ が定義されているとする。このとき、G の極大木のうち、弧の重みの総和が最小である極大木を最小木と呼ぶ。最小木を求めるアルゴリズムに Jarník-Prim のアルゴリズムがある。

$U \subseteq V$ をすでに極大木 $T \subseteq A$ の一部となった頂点の集合とする。探索の始点を $v_0 \in V$ とし、初期値は $U = \{v_0\}$、$T = \emptyset$ とすると、アルゴリズムは Algorithm 3.1 と記述できる。

$U = \{v_o\}$
$T = \emptyset$
while($U \neq V$){
　　U と $V \backslash U$ を結ぶ弧のうち重み最小のものを a とする;
　　a の $V \backslash U$ 側の頂点を w とする;
　　$U \leftarrow U \cup \{w\}$;//$U$ に w を追加
　　$T \leftarrow T \cup \{a\}$;//$T$ に a を追加
}

Algorithm 3.1: Jarník-Prim のアルゴリズム。//はコメント。

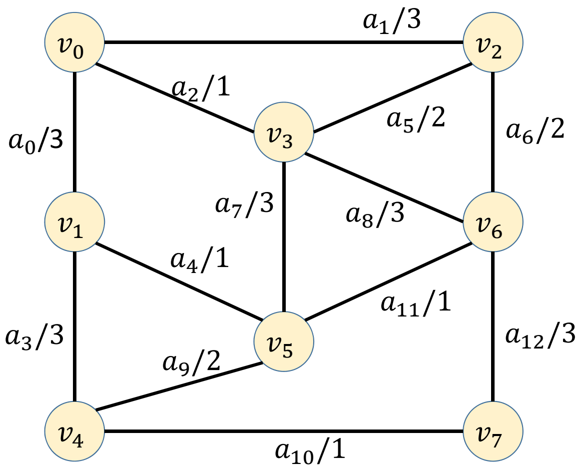


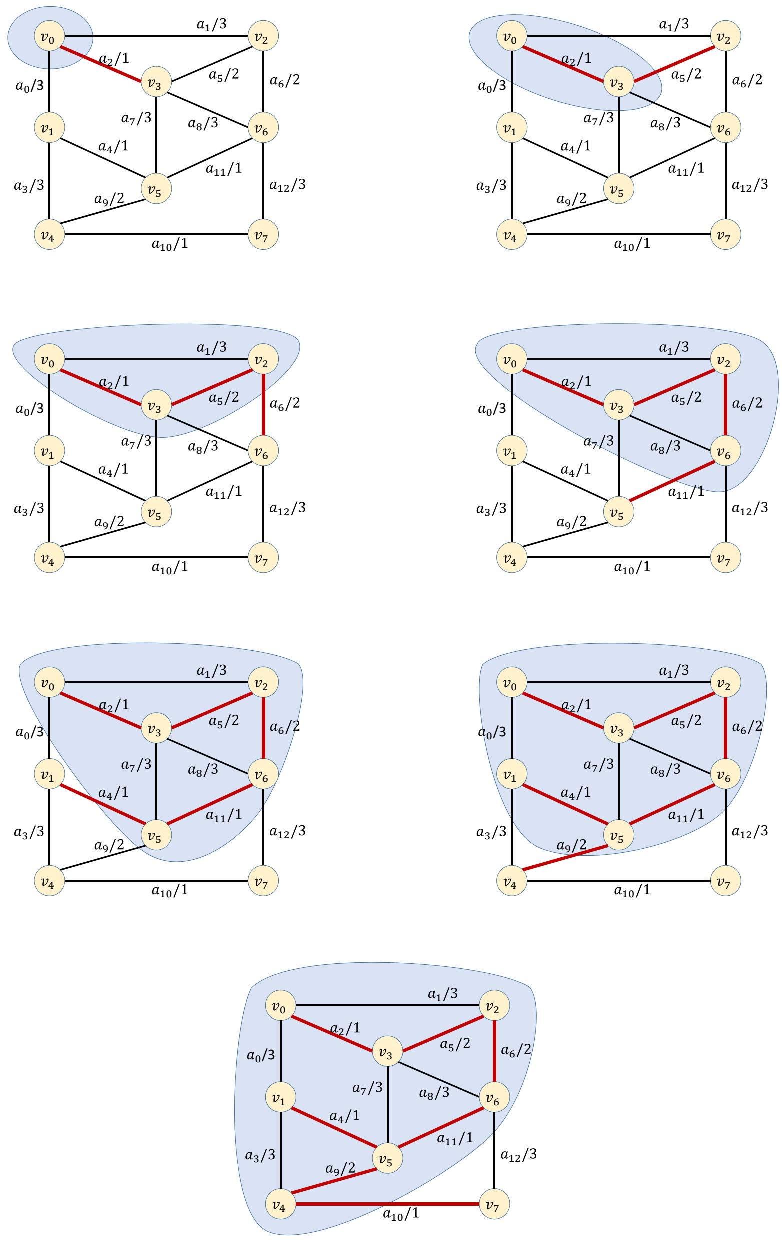
図 3: Algorithm 3.1 を用いる対象グラフ。各弧に添えられた数字はそれぞれの重みを表す。

3.1　最小木を求める

Algorithm 3.1 を図 3 のグラフに適用し、探索の始点を v_0 として、最小木を求めることを考える。各弧には弧のラベルとともに重さを示している。例えば $a_4/3$ は、弧のラベルが a_4 であり、$w(a_4) = 3$ であることを示している。最小木を求める途中経過が分かるように、集合 U に頂点が追加される順番を示しなさい。また、重みが同じ弧が複数ある場合には、弧の番号の小さいほうを

優先しなさい。**解答例**最小木が生成される様子を下図に示す。頂点の集合を囲む塗りつぶしが、U に相当する。U に頂点が追加される順番は以下のようになる。

$$v_0 \to v_3 \to v_2 \to v_6 \to v_5 \to v_1 \to v_4 \to v_7$$



3.2　最小木

得られた最小木を示しなさい。

解答例