「グラフと組合せ」試験問題　解答例

2011 年度

１数学的帰納法(20 点)

次の漸化式

$$P(1) = \frac{2}{3} \tag{1.1}$$

$$(k+2)P(k) = (k-1)P(k-1) \quad (k>1) \tag{1.2}$$

を満たす数列$P(k)$ $(k \geq 1)$は、以下の(1.3)式で表されることを、数学的帰納法を用いて証明しなさい。

$$P(k) = \frac{4}{(k+2)(k+1)k} \tag{1.3}$$

解答例

- $k=1$の場合：(1.3)式で$k=1$とすると

$$P(1) = \frac{4}{(1+2)(1+1)1} = \frac{2}{3}$$

となり、(1.1)式となる。

- ある$k-1 \geq 1$で、(1.3)式が成り立つとする。(1.4)式より、kでの表式を求める。

$$P(k) = \frac{k-1}{k+2} \times P(k-1)$$

ここで、右辺に、$k-1$に対応した(1.3)式の表式を代入する。

$$P(k) = \frac{k-1}{k+2} \times \frac{4}{(k+1)k(k-1)}$$
$$= \frac{4}{(k+2)(k+1)k}$$

つまり、kに対応した(1.3)式を得る。

- 以上より、任意の自然数$k \geq 1$に対して(1.3)式が漸化式(1.1)及び(1.2)の解であることが、数学的帰納法によって証明された。

２幅優先探索

単純な（孤立弧及び並列弧が無い）有向グラフ$G=(V,A)$に対する、幅優先探索(Breadth-First Search）のアルゴリズムを以下に示す。ただし、頂点$v \in V$を始点とする弧

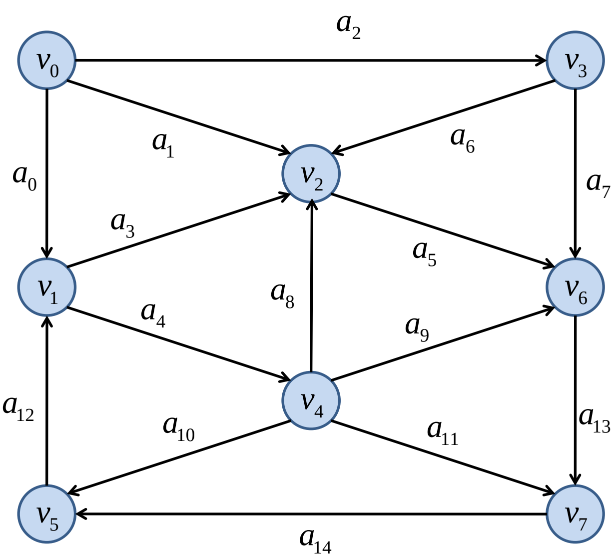
の集合を$\delta^+ v$、弧aの終点となる頂点を$\partial^- a$とする記法を用いている。$L \subseteq V$は既に探索した頂点の集合、Qは、探索すべき頂点の待ち行列とする。探索の始点となる頂点をrとする。

$L = \emptyset$

$Q = [r]$

while($Q \neq \emptyset$){
  $v = Q.poll$ //待ち行列Qの先頭の取出し
  forall($a \in \delta^+ v$){
    $w = \partial^- a$
    if($w \notin L \wedge w \notin Q$){
      $Q.append(w)$ // wをQに追加
    }
  }
  $L \leftarrow L \cup \{v\}$
}

このアルゴリズムを用いて、下図のグラフを、v_0を始点として幅優先探索を行う。



A) 探索の途中経過(20 点)

探索の途中経過を、注目している頂点v、その時のリストL及び待ち行列Qに保存され

ている頂点として以下のように示しなさい。

注目している頂点v	リストL	待ち行列Q
	$\varnothing$	$[v_0]$
v_0	$\{v_0\}$	$[v_1, v_2, v_3]$
v_1	$\{v_0, v_1\}$	$[v_2, v_3, v_4]$

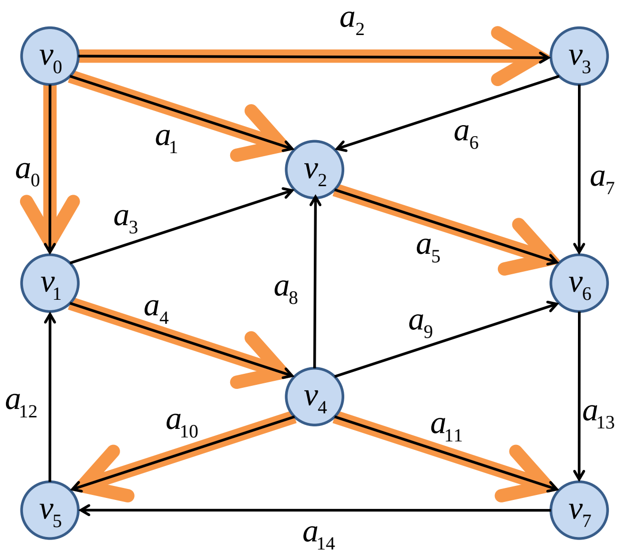
解答例

注目している頂点v	リストL	待ち行列Q
	$\varnothing$	$[v_0]$
v_0	$\{v_0\}$	$[v_1, v_2, v_3]$
v_1	$\{v_0, v_1\}$	$[v_2, v_3, v_4]$
v_2	$\{v_0, v_1, v_2\}$	$[v_3, v_4, v_6]$
v_3	$\{v_0, v_1, v_2, v_3\}$	$[v_4, v_6]$
v_4	$\{v_0, v_1, v_2, v_3, v_4\}$	$[v_6, v_5, v_7]$
v_6	$\{v_0, v_1, v_2, v_3, v_4, v_6\}$	$[v_5, v_7]$
v_5	$\{v_0, v_1, v_2, v_3, v_4, v_5, v_6\}$	$[v_7]$
v_7	$\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$	$\varnothing$

B)　極大木(10 点)

幅優先探索の結果として得られる極大木を示しなさい。

解答例



C)　最短距離(20 点)

弧に対応した頂点間の距離が全て 1 であると仮定すると、幅優先探索は始点からの最短距離を求める方法となっている。始点からある頂点vへの距離を$\phi(v)$とするとき、幅優先探索によって最短距離を求めることができるように、アルゴリズムを拡張しなさい。

解答例

```
L = ∅
Q = [r]
φ(r) = 0
while( Q ≠ ∅ ){
        v = Q.poll   //待ち行列 Q の先頭の取出し
        forall( a ∈ δ⁺v ){
                w = ∂⁻a
                if( w ∉ L ∧ w ∉ Q ){
                        Q.append(w)   // w を Q に追加
                        φ(w) = φ(v) + 1
                }
        }
        L ← L ∪ {v}
}
```

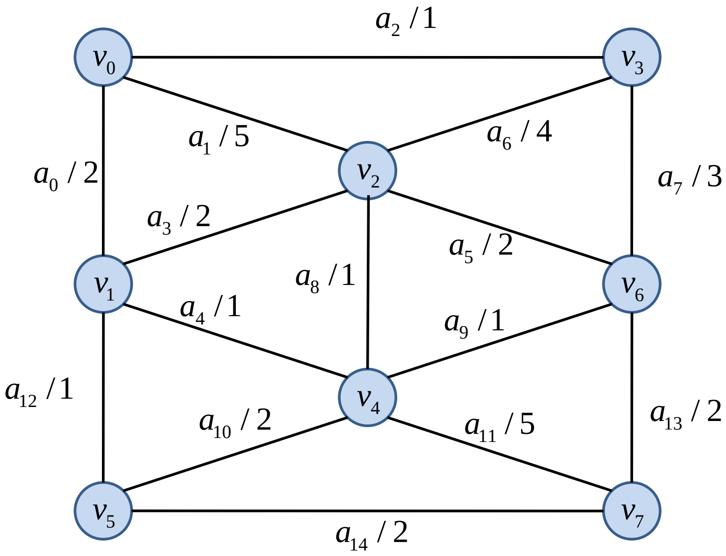
3 最小木を求めるJarník-Primアルゴリズム

単純な無向グラフ$G=(V,A)$において、各弧$a \in A$に正の重み$w(a)$が定義されているとする。このとき、ある頂点$v_0 \in V$を起点として、重みの和が最小となる極大木を求める方法の一つがJarník-Primアルゴリズムである。

$U \subseteq V$を、最小木の一部となった頂点の集合、$T \subseteq A$を、最小木とする。アルゴリズムは以下のように表される。

```
U = {v₀}
T = ∅
while ( U ≠ V ) {
        U と V \ U を結ぶ弧のうち、最小の重みのものを a とする
        a の V \ U 側の端点を w とする
        U ← U ∪ {w}
        T ← T ∪ {a}
}
```

A)　アルゴリズムの実行(20 点)



上のグラフについて、Jarník-Prim アルゴリズムにより最小木を求める際に、最小木に追加される弧の順序を示しなさい。始点は v_0 とする。図中の弧のラベルの右の数字は重みを表している。

解答例

$a_2, a_0, a_4, a_8, a_9, a_{12}, a_{13}$

B)　最小木(10 点)

　前問で得られた最小木を示しなさい。

解答例