Árvores Filogenéticas

Zanoni Dias

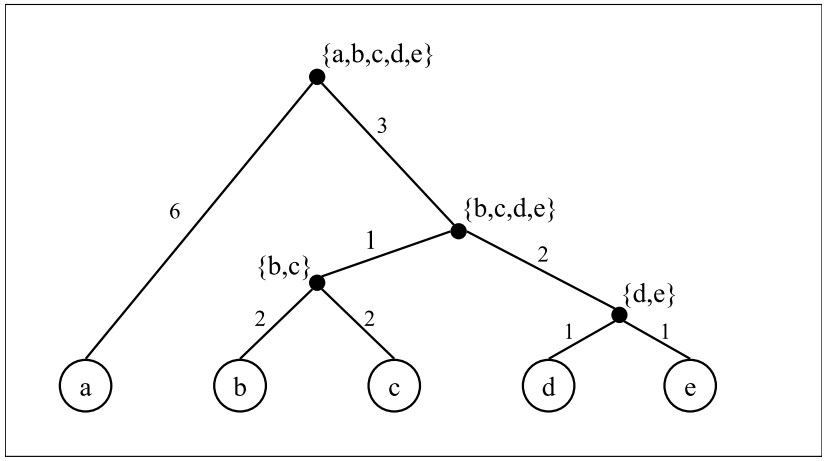
Instituto de Computação – Unicamp

16 de junho de 2009

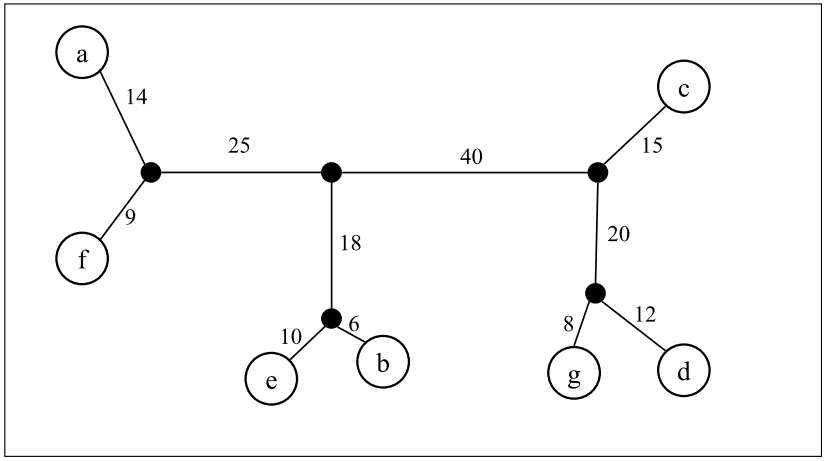
Árvores Filogenéticas

- Árvores filogenéticas representam abordagem mais comum usada para reconstruir a relação entre objetos biológicos.
- Cada folha da árvore denota um dos objetos biológicos, enquanto os nós internos representam ancestrais hipotéticos.
- A distância entre os objetos na árvore pode servir com uma medida do grau de relação entre os objetos.
- Em relação a árvores filogenéticas, há dois interesses principais:
  - Obter a topologia da árvore, ou seja, a forma como os nós internos se conectam uns com os outros e com as folhas.
  - Obter as distâncias entre todos os nós da árvore.
- Em relação a raiz de uma árvore filogenética, temos dois casos:
  - Nas árvore com raiz (ou enraizada), a raiz representa o ancestral comum a todos os nós da árvore.
  - Nem sempre temos informações suficientes para determinar o ancestral comum a todos os nós. Neste caso, constroi-se uma árvore sem raiz.

Árvore Filogenética com Raiz



Árvore Filogenética sem Raiz



Dados para Construção de Árvores Filogenéticas

- Os tipos de informações utilizadas para reconstrução filogenética são, normalmente, divididos em três categorias:
  - Informação comparativa numérica, chamada Matriz de Distância entre os pares de objetos.
  - Características discretas, tais como cor da pele, número de dedos, presença de asas, presença de um sítio de restrição, presença de um SNP, etc. Cada característica possui um número finito de estados (valores distintos que a característica pode assumir). Neste caso a informação é organizada numa matriz chamada Matriz de Estados das Características.
  - Características contínuas, tais como altura na fase adulta, peso no nascimento, tamanho do genoma, etc. Cada característica pode possuir um número infinito de estados. Neste caso a informação também pode ser organizada numa Matriz de Estados das Características.

Árvores Aditivas

Definição

Seja $\mathcal{A}$ um conjunto de objetos e $\delta : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ uma função. Então δ é uma métrica para $\mathcal{A}$ se satisfaz as seguintes propriedades:

- *Para todo par $a, b \in \mathcal{A}$, $\delta(a, b) = 0$ se e somente se $a = b$.*
- *Para todo par $a, b \in \mathcal{A}$, $\delta(a, b) = \delta(b, a)$ (simetria).*
- *Para toda trinca $a, b, c \in \mathcal{A}$, $\delta(a, b) \leq \delta(a, c) + \delta(c, b)$ (desigualdade triangular).*

Definição

Seja $\mathcal{A}$ um conjunto de objetos e $\delta : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ uma medida de distância métrica para $\mathcal{A}$. Seja $T = (V, E, d)$ uma árvore ponderada tal que $\mathcal{A} \subseteq V$. Seja $dist(x, y)$ a distância entre dois vértice quaisquer x e y de em T, calculada como a soma dos pesos das arestas do caminho entre x e y. A árvore T é chamada aditiva para $\mathcal{A}$ e δ se e somente se, para todo $a, b \in \mathcal{A}$, $dist(a, b) = \delta(a, b)$.

Como Verificar se uma Matriz de Distâncias é Aditiva

Teorema

Seja $\mathcal{A}$ um conjunto de objetos e $\delta : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ uma medida de distância métrica para $\mathcal{A}$. A métrica δ é dita aditiva para $\mathcal{A}$, ou seja admite uma árvore aditiva, se e somente se para todo conjunto de 4 elementos $i, j, k, l \in \mathcal{A}$, temos que:

- *ou $\delta(i,j) + \delta(k,l) = \delta(i,k) + \delta(j,l) \geq \delta(i,l) + \delta(j,k)$,*
- *ou $\delta(i,l) + \delta(k,j) = \delta(i,k) + \delta(j,l) \geq \delta(i,j) + \delta(k,l)$,*
- *ou $\delta(i,j) + \delta(k,l) = \delta(i,l) + \delta(k,j) \geq \delta(i,k) + \delta(j,l)$.*

- *Teorema provado, independentemente, por Peter Buneman (1971) e Annete Dobson (1974).*

Como Construir uma Árvore Aditiva

Lema

Seja $\mathcal{A} = \{x, y, z\}$ um conjunto de objetos e $\delta : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ uma distância métrica aditiva para $\mathcal{A}$. Logo podemos construir uma árvore aditiva $T = (V, E, d)$, com $V = \{x, y, z, c\}$, $E = \{\{x, c\}, \{y, c\}, \{z, c\}\}$, com pesos para as arestas dados pelas seguintes fórmulas:

$$d(x, c) = \frac{\delta(x, y) + \delta(x, z) - \delta(y, z)}{2}$$

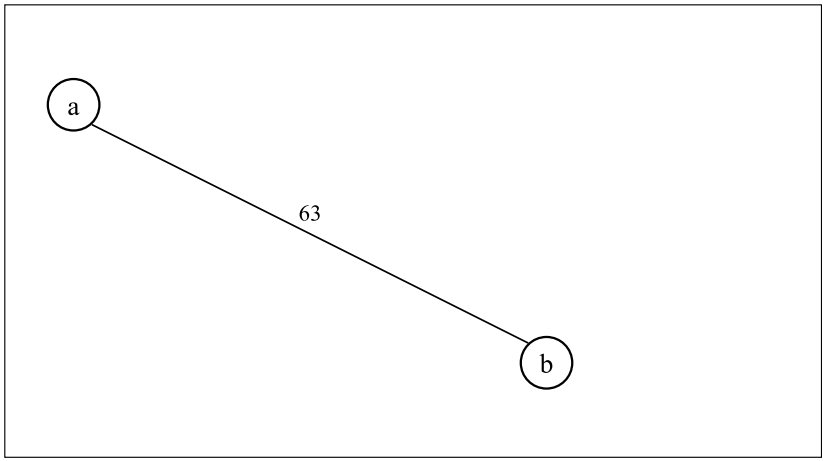
$$d(y, c) = \frac{\delta(x, y) + \delta(y, z) - \delta(x, z)}{2}$$

$$d(z, c) = \frac{\delta(x, z) + \delta(y, z) - \delta(x, y)}{2}$$

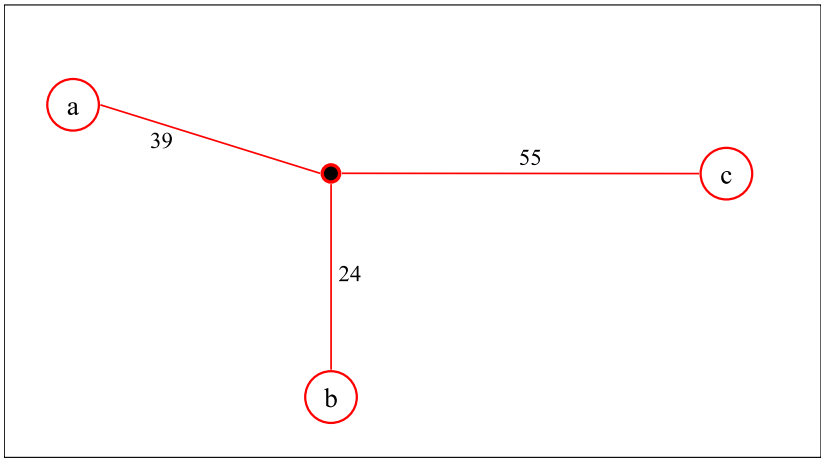
Árvore Aditiva

	a	b	c	d	e	f	g
a	0	63	94	111	67	23	107
b	63	0	79	96	16	58	92
c	94	79	0	47	83	89	43
d	111	96	47	0	100	106	20
e	67	16	83	100	0	62	96
f	23	58	89	106	62	0	102
g	107	92	43	20	96	102	0

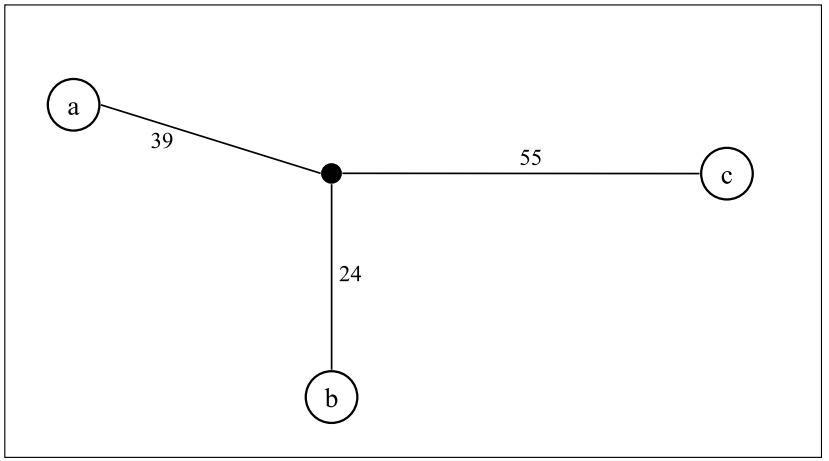
Árvore Aditiva



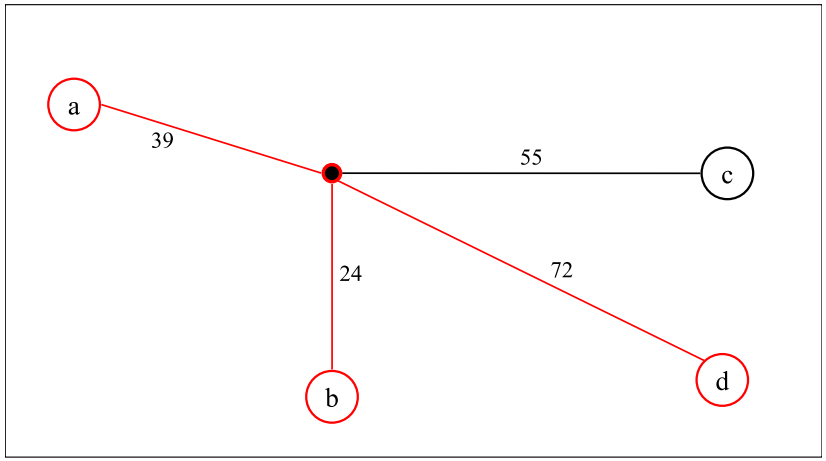
Árvore Aditiva



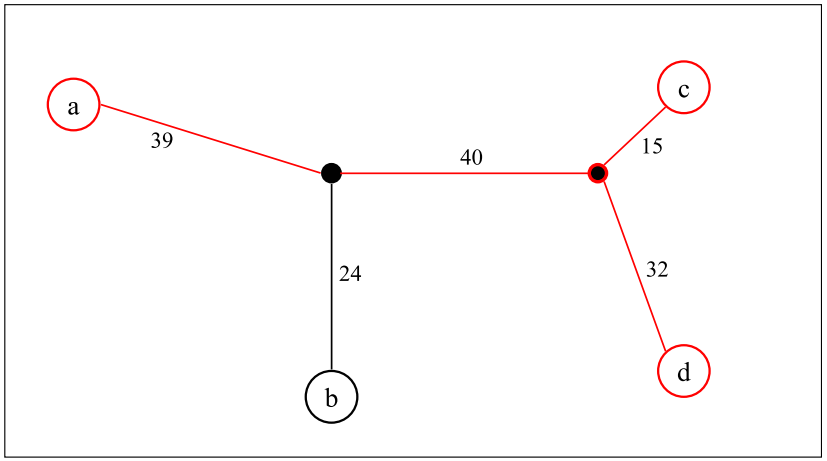
Árvore Aditiva



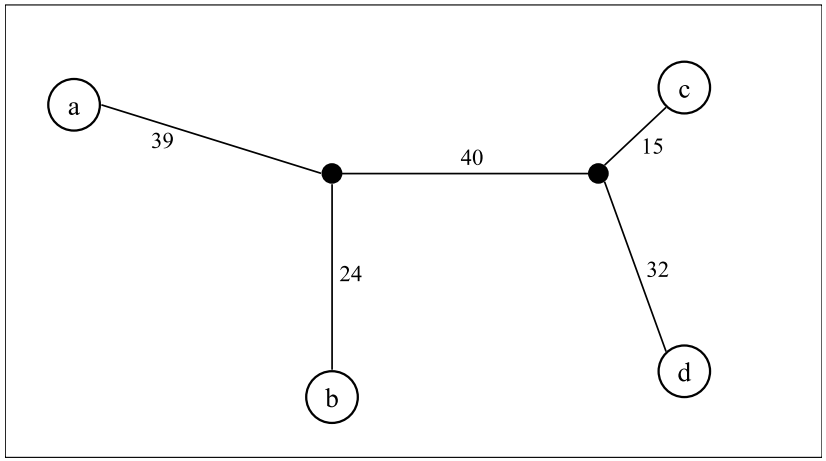
Árvore Aditiva



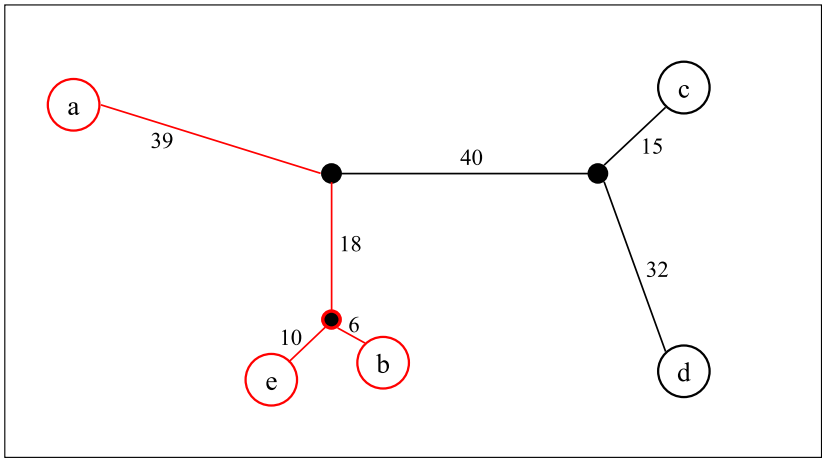
Árvore Aditiva



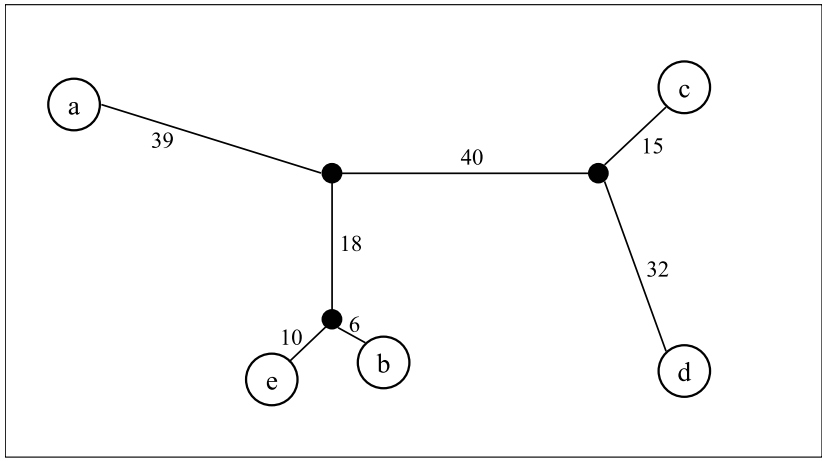
Árvore Aditiva



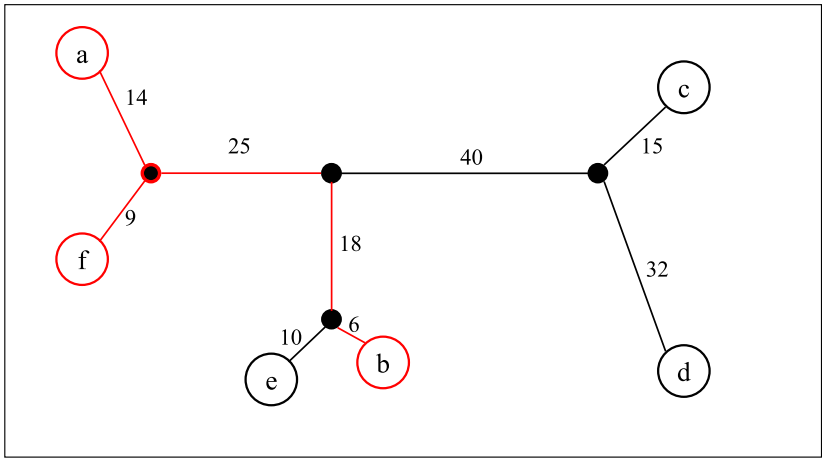
Árvore Aditiva



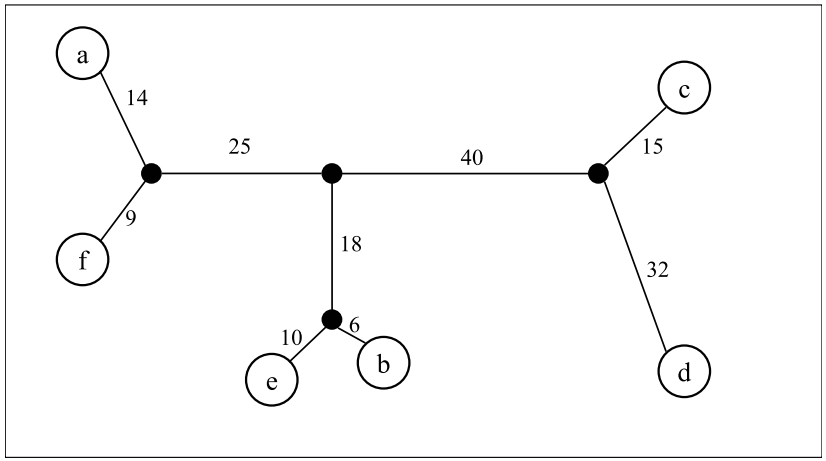
Árvore Aditiva



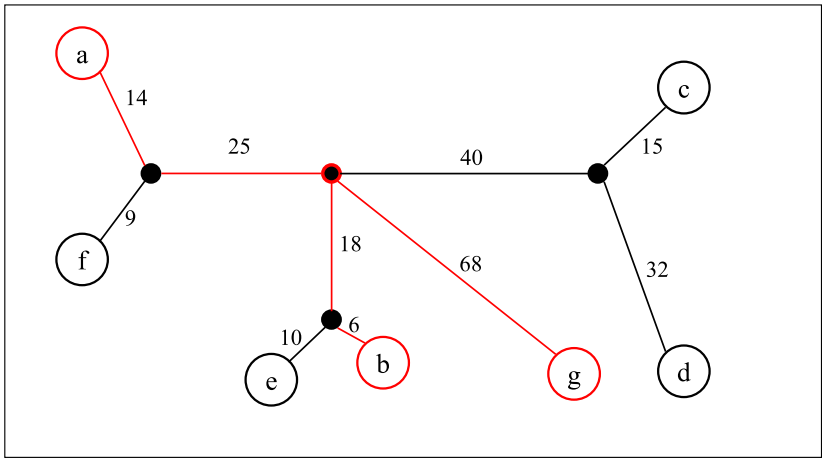
Árvore Aditiva



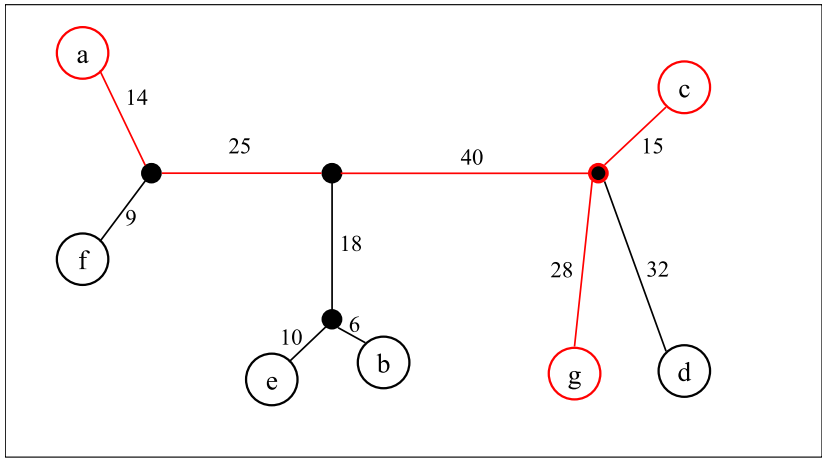
Árvore Aditiva



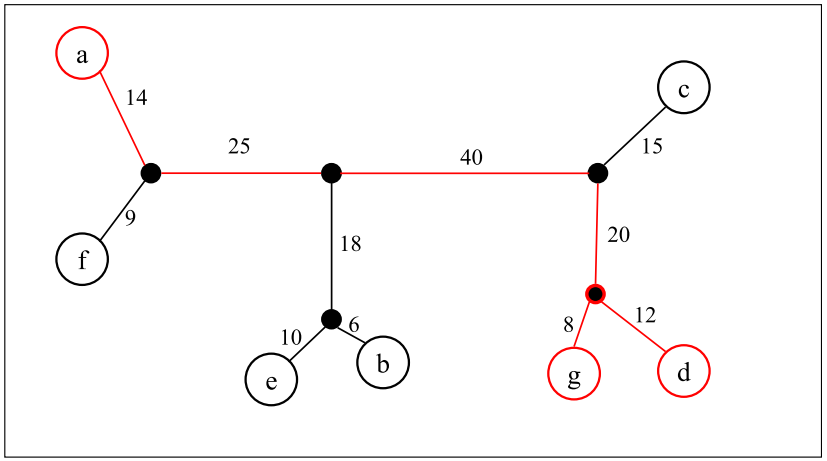
Árvore Aditiva



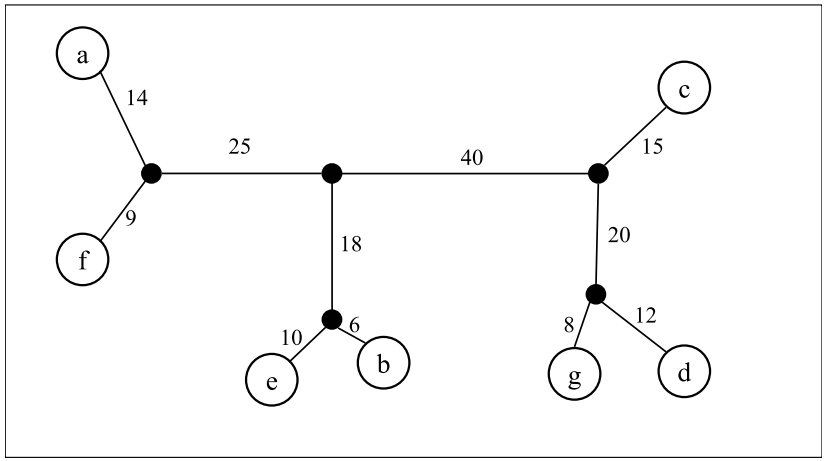
Árvore Aditiva



Árvore Aditiva



Árvore Aditiva



Algoritmo para Construção de Árvores Aditivas

- A partir de uma árvore formada por 3 vértices (quaisquer), adicione novos vértices, um a um, seguindo os seguintes passos:
  1. Escolha dois vértices quaisquer da árvore previamente construída (chame estes vértices de x e y).
  2. Calcule onde o novo vértice z deverá ser incluído, em relação ao caminho entre x e y.
  3. Se a inserção do novo vértice gerar um novo vértice interno c, entre os vértices c_1 e c_2, remova a aresta (c_1, c_2), insira os vértices c e z e as arestas (c_1, c), (c, c_2) e (c, z).
  4. Caso contrário, se existir um vértice y' da árvore previamente construída (e ainda não descartado na inserção do vértice corrente), chame-o de y e volte ao passo 2.
  5. Caso contrário, insira o vértice z e aresta (c, z), onde c é o nó interno do caminho entre x e y onde z deve ser incluído.
- Complexidade (pior caso): $\sum_{k=4}^{n}(k-2)(k-2) = \Theta(n^3)$.
- Algoritmo proposto por Waterman, Smith, Singh e Beyer, em 1977.

Árvores Aditivas Compactas

Definição

Uma árvore aditiva $T = (V, E, d)$ é chamada compacta se $\mathcal{A} = V$.

Teorema

Seja $G(V, E)$ o grafo completo onde os vértices representam os objetos de $\mathcal{A}$ e as arestas representam as distâncias métricas entre todos os pares de objetos. O grafo $G(V, E)$ é chamado Grafo de Distâncias. Se existe uma árvore compacta aditiva $T = (V, E', d)$, com $E' \subseteq E$, para $\mathcal{A}$ com respeito a δ, então T é a única Árvore Geradora Mínima do grafo $G(V, E)$.

- Como obter uma Árvore Aditiva Compacta (caso ela exista):
  - Execute o algoritmo de Prim para Árvore Geradora Mínima: $\Theta(n^2)$.
  - Usando Ordenação Topológica, calcule a distância entre todos os pares de vértices da Árvore Geradora Mínima: $n \times \Theta(n) = \Theta(n^2)$.
  - Para cada par de vértice $i, j \in V$, teste se $dist(i, j) = \delta(i, j)$: $\Theta(n^2)$.
  - Complexidade: $\Theta(n^2)$.

Árvores Filogenéticas para Matrizes não Aditivas

- Em muitos casos práticos a matriz distância não é aditiva.
- Nestes casos, estamos interessados em encontrar a melhor árvore filogenética em relação a matriz de distâncias.
- Existem muitas formas possíveis de definir "a melhor árvore filogenética" em relação a uma matriz de distâncias, por exemplo, a árvore que satizfaça a seguintes expressão:

$$min \sum_{i,j \in \mathcal{A}} |dist(i, j) - \delta(i, j)|$$

- William Day, em 1987, provou que o problema de encontrar a melhor árvore filogenética, sob várias medidas diferentes, é um problema *NP*-Completo.

Distância Ultramétrica

Definição

Seja $\mathcal{A}$ um conjunto de objetos e $\delta : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ uma métrica para $\mathcal{A}$. Então δ é uma ultramétrica para $\mathcal{A}$ se satisfaz a seguinte condição:

- *Para toda trinca $a, b, c \in \mathcal{A}$, ou $\delta(a, b) \leq \delta(a, c) = \delta(c, b)$ ou $\delta(a, c) \leq \delta(a, b) = \delta(b, c)$ ou $\delta(b, c) \leq \delta(b, a) = \delta(a, c)$.*

Lema

Uma matriz de distâncias M é ultramétrica se e somente se no grafo completo G correspondente, a aresta de maior peso, em qualquer ciclo, não é única.

Observação

Toda distância ultramétrica é uma distância aditiva.

Árvores Ultramétricas

Definição

Seja $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ um conjunto de objetos. Uma árvore ponderada $T = (V, E, d)$ com raiz r e função de peso associada às arestas $d : E \to \mathbb{R}^+$ é uma árvore ultramétrica para $\mathcal{A}$ se satisfaz as seguintes condições:

- *T é uma árvore binária, ou seja, cada vértice interno de T possui exatamente dois filhos.*
- *T possui exatamente n folhas, rotuladas com $\{a_1, a_2, \ldots, a_n\}$.*
- *A soma dos pesos das arestas de qualquer caminho da raiz r a qualquer folha de T é sempre o mesmo.*

Árvores Ultramétricas

Teorema

Seja $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ um conjunto de objetos e $\delta : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ uma função de comparação para os objetos de $\mathcal{A}$, logo existe uma árvore ultramétrica para $\mathcal{A}$ se e somente se δ for ultramétrica.

- Dado um conjunto de objetos $\mathcal{A}$ e uma função ultramétrica δ para $\mathcal{A}$, o algoritmo `UPGMA` (*Unweighted Pair Group Method with Arithmetic Mean*), proposto por Robert Sokal e Charles Michener em 1958, calcula a árvore ultramétrica para $\mathcal{A}$ em $O(n^3)$.

UPGMA

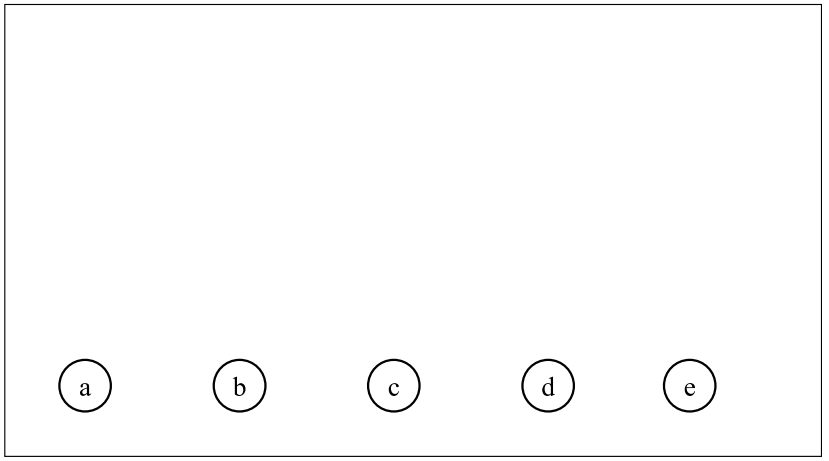
Algoritmo 1: UPGMA

Input: $\mathcal{A}, n, \delta$
$X \leftarrow \{\{a_i\}, \{a_2\}, \ldots, \{a_n\}\}$
for *all* $i, j \in [1..n]$ **do** $dist(\{a_i\}, \{a_j\}) = \delta(a_i, a_j)$
for *all* $i \in [1..n]$ **do** $height(\{a_i\}) \leftarrow 0$
$V \leftarrow X$
$E \leftarrow \emptyset$
while $|X| \geq 2$ **do**
  $min \leftarrow 0$
  for *all* $x_i, x_j \in X$ **do** **if** $x_i \neq x_j$ *and* $dist(x_i, x_j) < min$ **then**
    $min \leftarrow dist(x_i, x_j)$
    $C_1 \leftarrow x_i$; $C_2 \leftarrow x_j$; $D \leftarrow C_1 \cup C_2$
  end
  $X \leftarrow (X - \{C_1, C_2\}) \cup \{D\}$
  for *all* $C \in X$ **do** $dist(D, C) \leftarrow dist(C, D) \leftarrow (dist(C_1, C) + dist(C_2, C))/2$
  $V \leftarrow V \cup \{D\}$
  $E \leftarrow E \cup \{(D, C_1), (D, C_2)\}$
  $height(D) \leftarrow (dist(C_1, C_2))/2$
  $d(D, C_1) \leftarrow height(D) - height(C_1)$
  $d(D, C_2) \leftarrow height(D) - height(C_2)$
end
return $T = (V, E, d)$

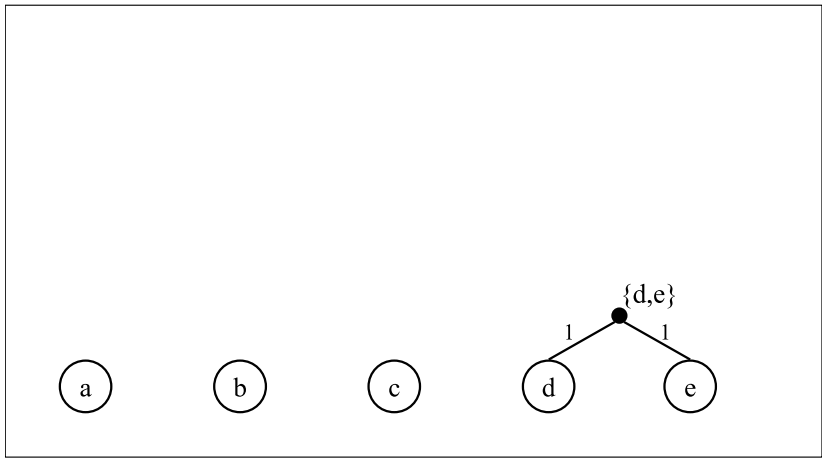
Árvore Ultramétrica

	a	b	c	d	e
a	0	12	12	12	12
b	12	0	4	6	6
c	12	4	0	6	6
d	12	6	6	0	2
e	12	6	6	2	0

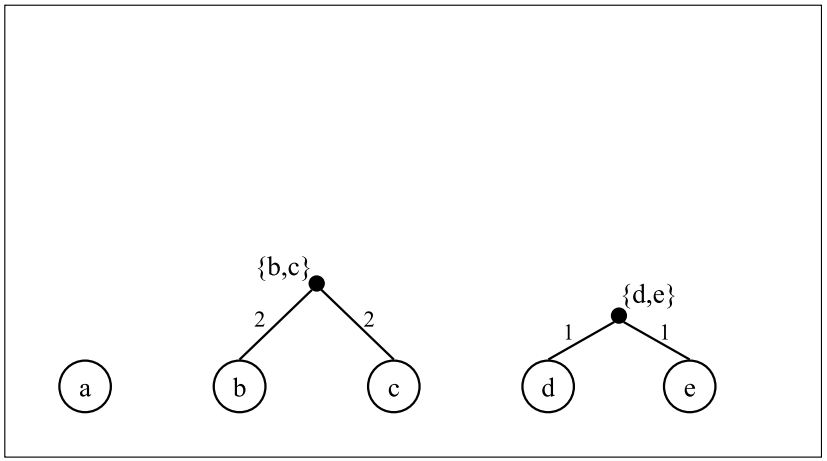
Árvore Ultramétrica



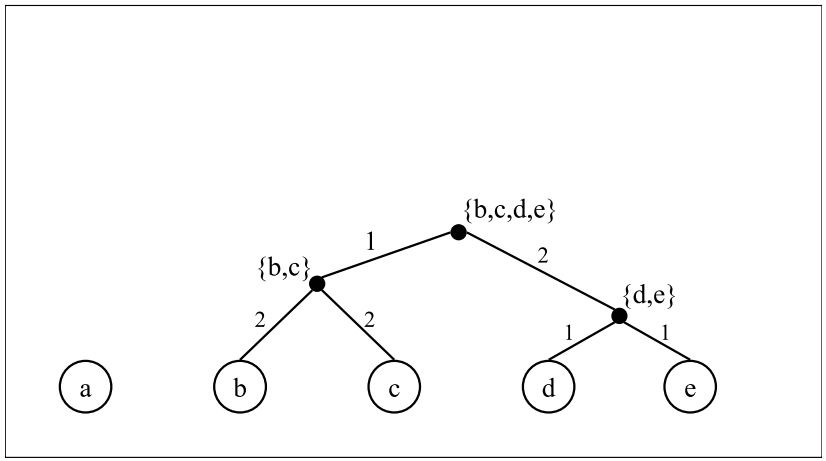
Árvore Ultramétrica



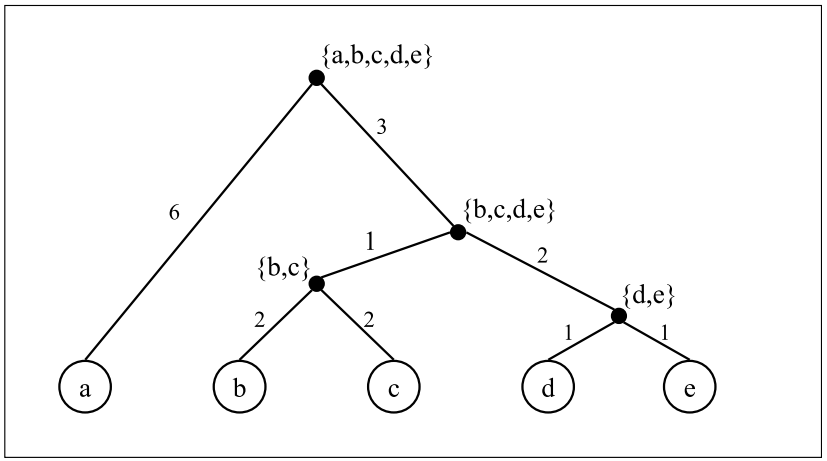
Árvore Ultramétrica



Árvore Ultramétrica



Árvore Ultramétrica



Definição

Seja M^l e M^h duas matrizes de distâncias entre os objetos $\mathcal{A}$, contendo, respectivamente, limites inferiores e superiores para as distâncias entre os pares de objetos de $\mathcal{A}$. Ou seja, para todo par $i, j \in \mathcal{A}$, temos que:

$$M^l[i,j] \leq \delta(i,j) \leq M^h[i,j]$$

Definição

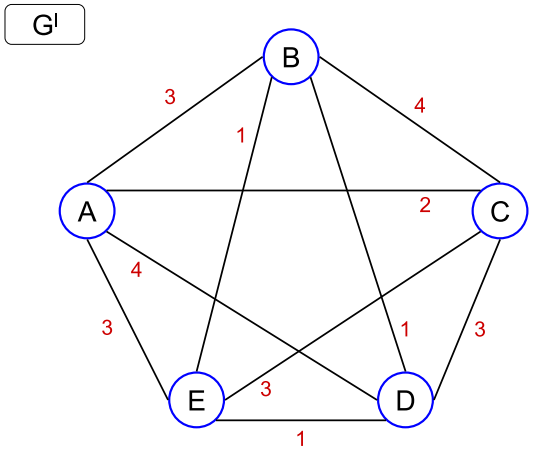
Seja T uma árvore geradora mínima para o grafo G^h construído a partir da matriz M^h. O corte mais pesado (cut-weight) de uma aresta e de T é dado por:

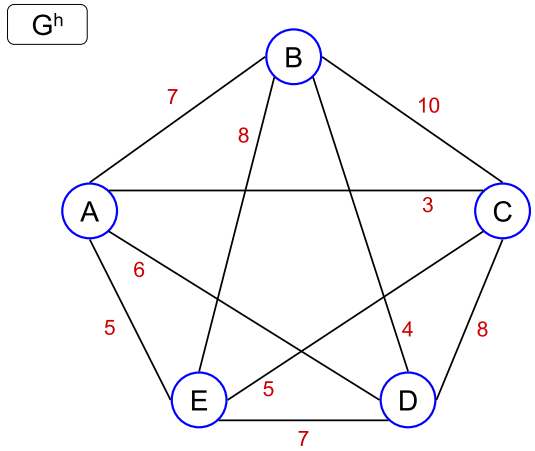
$$CW(e) = max\{M^l[a,b] | e = (a,b)_{max}\}$$

onde $(a,b)_{max}$ é a aresta mais pesada do caminho entre os vértices a e b na árvore geradora mínima T.

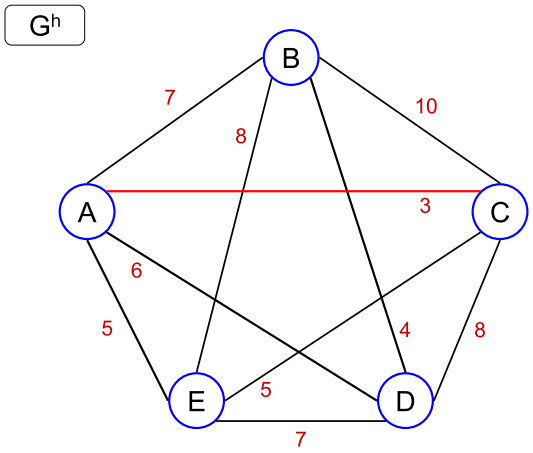
Algoritmo 2: Sandwich Ultrametric

Input: $\mathcal{A}, T, CW$
for *all* $i \in \mathcal{A}$ **do**
  MakeSet(i)
  CreateNode(i)
  $height[i] \leftarrow 0$
end
Sort edges of T in nondecreasing order of cut-weights
for *all edge* $e = (a,b) \in T$ *in that order* **do**
  $A \leftarrow$ FindSet(a); $B \leftarrow$ FindSet(b)
  if $A \neq B$ **then**
    $u_a \leftarrow$ root of the tree that contains a
    $u_b \leftarrow$ root of the tree that contains b
    CreateNode(U)
    $U.left \leftarrow u_a$
    $U.right \leftarrow u_b$
    $height[U] \leftarrow CW[e]/2$
    $d(U, u_a) \leftarrow height(U) - height(u_a)$
    $d(U, u_b) \leftarrow height(U) - height(u_b)$
    Union(A,B)
  end
end
return U

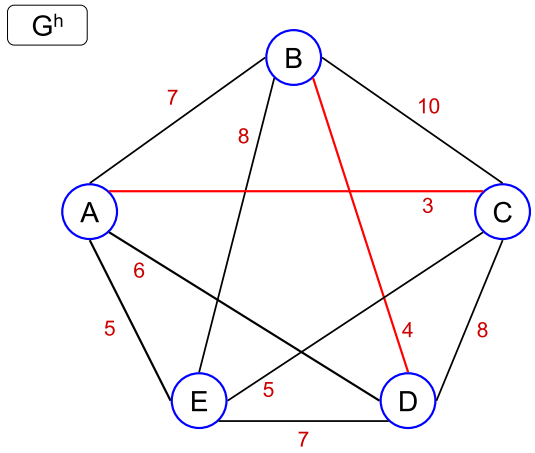




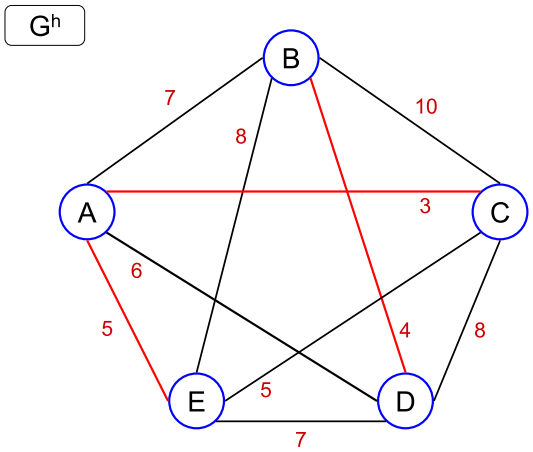
Sanduíche Ultramétrico



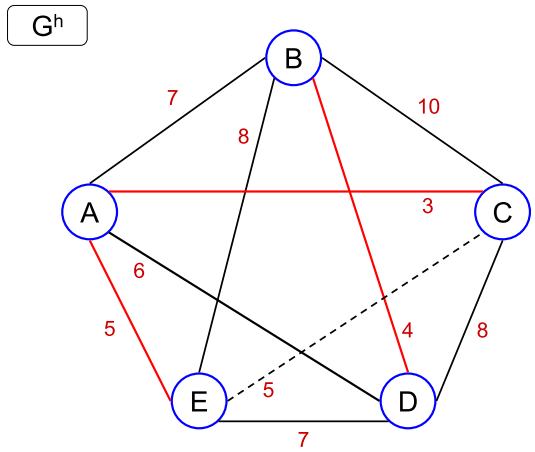
Sanduíche Ultramétrico



Sanduíche Ultramétrico

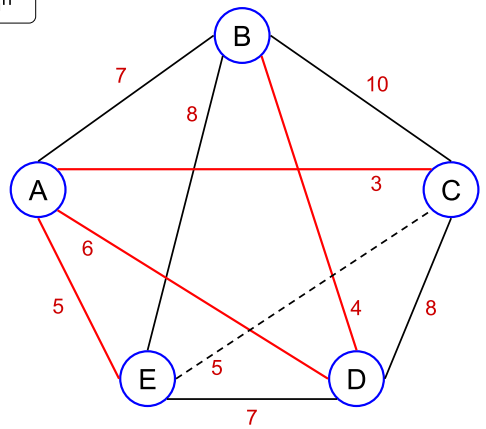


Sanduíche Ultramétrico



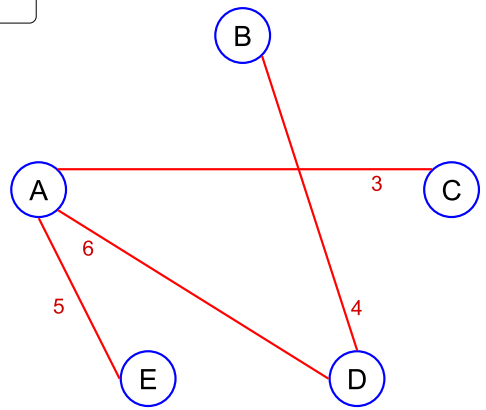
Sanduíche Ultramétrico

G^h



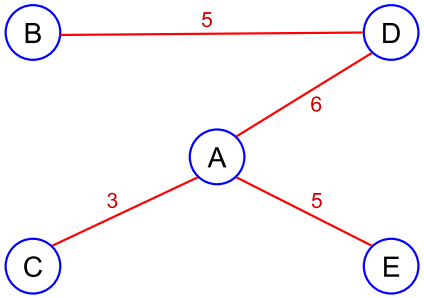
Sanduíche Ultramétrico

T



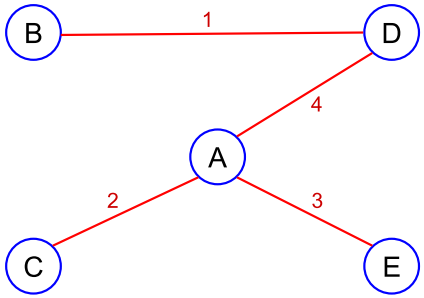
Sanduíche Ultramétrico

T

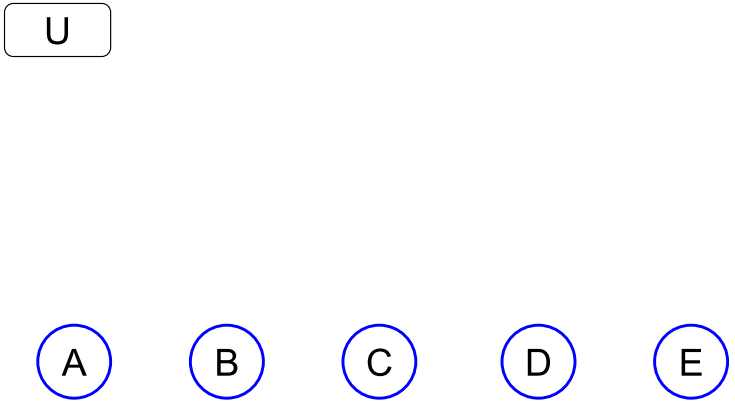


Sanduíche Ultramétrico

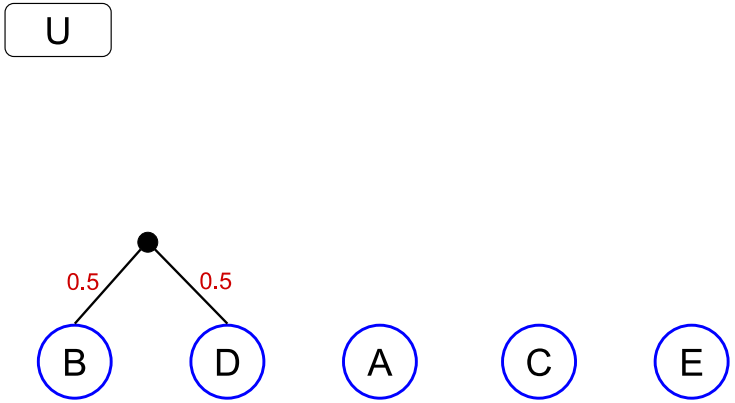
CW



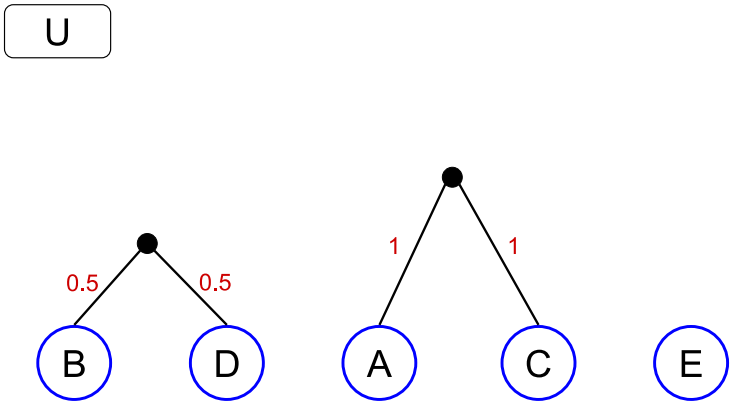
Sanduíche Ultramétrico



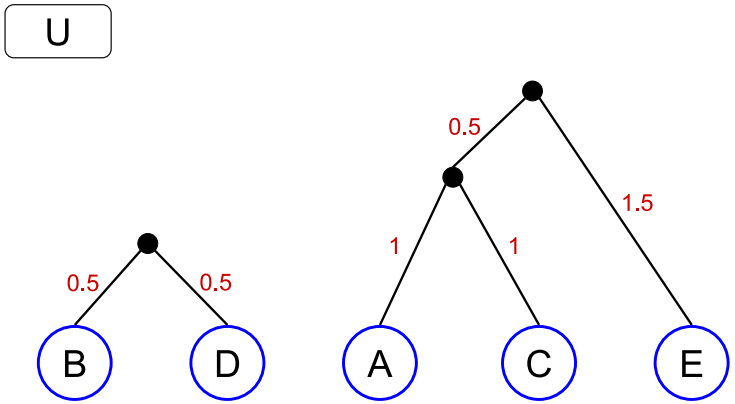
Sanduíche Ultramétrico



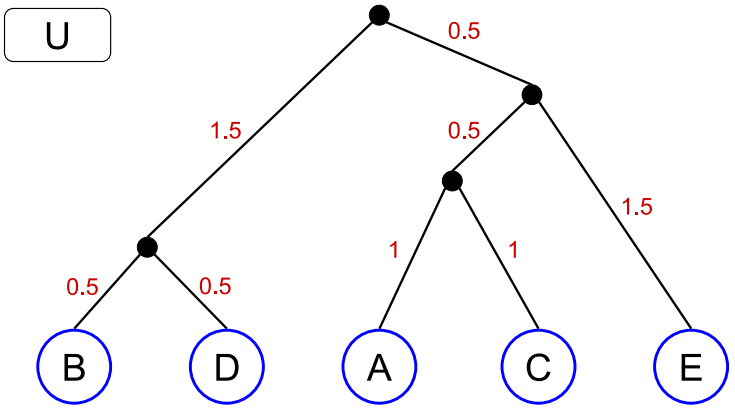
Sanduíche Ultramétrico



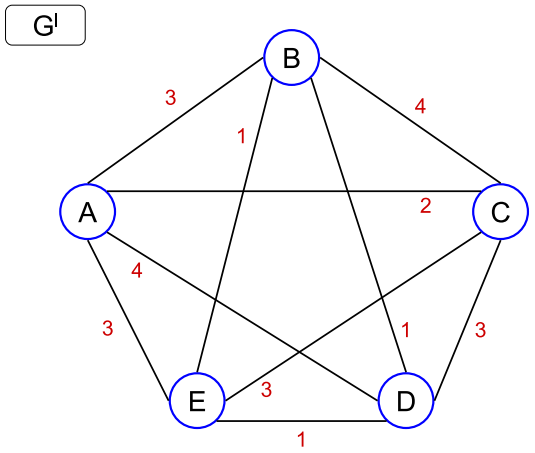
Sanduíche Ultramétrico



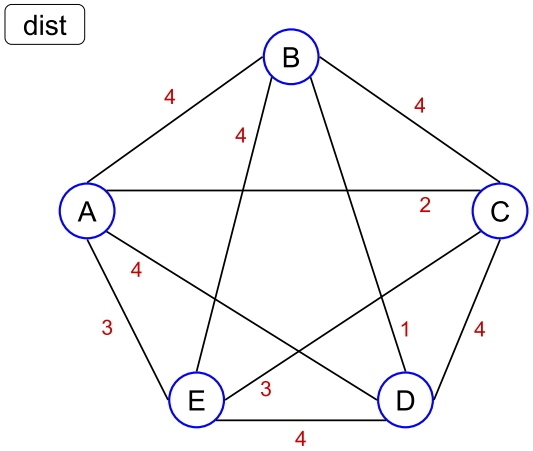
Sanduíche Ultramétrico



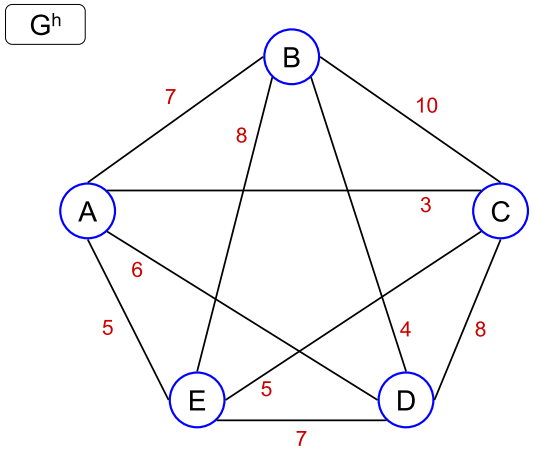
Sanduíche Ultramétrico



Sanduíche Ultramétrico



Sanduíche Ultramétrico



Sanduíche Ultramétrico

- Complexidade:
  - Construção da Árvore Geradora Mínima T de G^h: $\Theta(n^2)$, usando o algoritmo de Prim.
  - Cálculo de $CW[e]$, para toda aresta e de T:
    - Segundo a definição de CW: $\Theta(n) \times O(n^2) = O(n^3)$.
    - Construindo uma árvore binária auxiliar R, onde os objetos de $\mathcal{A}$ são folhas e os nós internos são as arestas de T, de tal forma que para cada par de objetos de $\mathcal{A}$, o nó interno que é o ancestral comum mais próximo de a e b contém a aresta $(a,b)_{max}$: $\Theta(n \log n) + \Theta(n^2) = \Theta(n^2)$.
  - Algoritmo `Sandwich Ultrametric`: $\Theta(n(\alpha(n) + \log n)) = \Theta(n \log n)$.
  - Total: $\Theta(n^2)$.
- Algoritmo proposto por Martin Farach, Sampath Kannan e Tandy Warnow, em 1993. Também provaram que o problema de obter uma árvore aditiva (não necessariamente ultramétrica) que satizfaça as restrições de "sanduíche" entre duas matrizes de distância é um problema $\mathcal{NP}$-Completo.

Existência de uma Árvore Ultramétrica

Teorema

Existe uma árvore ultramétrica U que respeita a restrição:

$$M^l[i,j] \leq dist(i,j) \leq M^h[i,j]$$

para todos os pares de objetos de $i,j \in \mathcal{A}$, se e somente se para todos os pares de objetos de $i,j \in \mathcal{A}$ é verdade que:

$$M^l[i,j] \leq d((i,j)_{max}).$$

Características com Estados Binários

- Todas as característica só tem dois estados possíveis:
  - 0: ausente.
  - 1: presente.
- Todas as características são independentes entre si.
- Não há nenhuma característica ausente ou presente em todos os objetos.
- Não existem dois ou mais objetos com todas as características no mesmo estado.
- Todas as características evoluem do estado 0 para o estado 1. Após alcançar o estado 1, uma característica nunca retorna ao estado 0.
- A raíz da árvore filogenética representará o ancestral com todas as características ausentes (estado 0 para todas as características).

Filogenia Perfeita para Características com Estados Binários

Definição

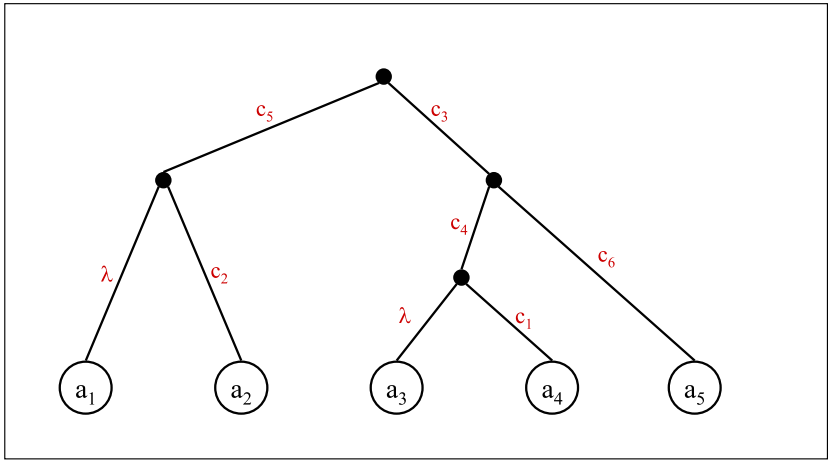
Seja $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ um conjunto de objetos, $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ um conjunto de características binárias e M uma Matriz de Estados de Caracterísicas para $\mathcal{A}$ e $\mathcal{C}$. Uma árvore filogenética perfeira para M é uma árvore $T = (V, E, d)$ com exatamente n folhas satizfazendo as seguintes condições:

- *As folhas de T correspondem aos objetos de $\mathcal{A}$.*
- *As arestas são rotuladas de acordo com a função $d : E \to \mathcal{C} \cup \{\lambda\}$, onde λ representa o rótulo vazio.*
- *Cada uma das características de $\mathcal{C}$ é atribuída a exatamente uma aresta de T.*
- *Para cada objeto a_i, o conjunto de rótulos do caminho de a_i em T até a raiz corresponde exatamente as características presentes em a_i.*

Filogenia Perfeita para Características com Estados Binários

	c_1	c_2	c_3	c_4	c_5	c_6
a_1	0	0	0	0	1	0
a_2	0	1	0	0	1	0
a_3	0	0	1	1	0	0
a_4	1	0	1	1	0	0
a_5	0	0	1	0	0	1

Filogenia Perfeita para Características com Estados Binários



Existência de Filogenia Perfeita

Definição

Para cada característica c_i de $\mathcal{C}$ seja $\mathcal{A}_i$ o conjunto de objetos de $\mathcal{A}$ tal que o estado da característica c_i seja igual a 1.

- Exemplos: $\mathcal{A}_3 = \{a_3, a_4, a_5\}$, $\mathcal{A}_5 = \{a_1, a_2\}$.

Lema

Uma matriz binária M admite uma filogenia perfeira se e somente se para cada par de características c_i e c_j os conjuntos $\mathcal{A}_i$ e $\mathcal{A}_j$ ou são disjuntos ou um deles contém o outro.

Construção de Filogenia Perfeita

Algoritmo 3: Perfect Phylogenetic Tree

Input: $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$, M
Sort the columns of M in nonincreasing order of numbers of ones
$V \leftarrow \{\mathcal{A}\}$
$E \leftarrow \emptyset$
for *all* $j \in [1..m]$ **do**
  Search for the vertex $X \in V$ representing the smallest superset of $\mathcal{A}_j$
  $V \leftarrow V \cup \{\mathcal{A}_j\}$
  $E \leftarrow E \cup \{(X, \mathcal{A}_j)\}$
  $d(X, \mathcal{A}_j) \leftarrow c_j$
end
for *all* $i \in [1..n]$ **do**
  if $a_i \notin V$ **then**
    Search for the vertex $X \in V$ representing the smallest set containing a_i
    $V \leftarrow V \cup \{a_i\}$
    $E \leftarrow E \cup \{(X, \{a_i\})\}$
    $d(X, \{a_i\}) \leftarrow \lambda$
  end
end
return $T = (V, E, d)$

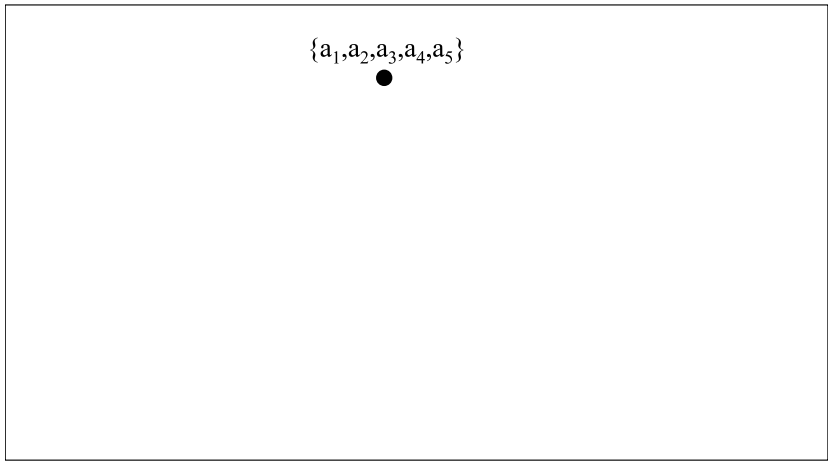
Construção de Filogenia Perfeita

	c_1	c_2	c_3	c_4	c_5	c_6
a_1	0	0	0	0	1	0
a_2	0	1	0	0	1	0
a_3	0	0	1	1	0	0
a_4	1	0	1	1	0	0
a_5	0	0	1	0	0	1

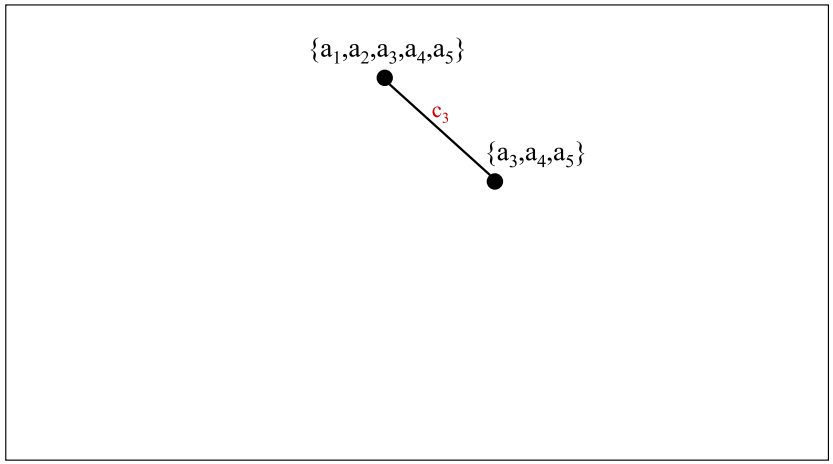
Construção de Filogenia Perfeita

	c_3	c_4	c_5	c_1	c_2	c_6
a_1	0	0	1	0	0	0
a_2	0	0	1	0	1	0
a_3	1	1	0	0	0	0
a_4	1	1	0	1	0	0
a_5	1	0	0	0	0	1

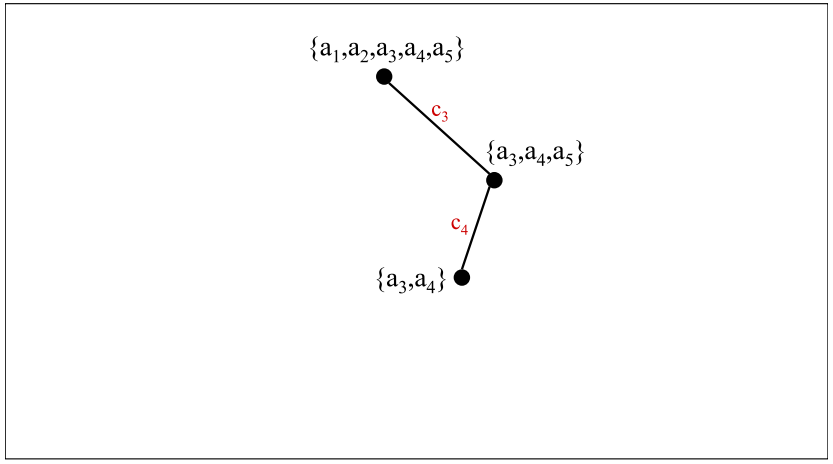
Construção de Filogenia Perfeita



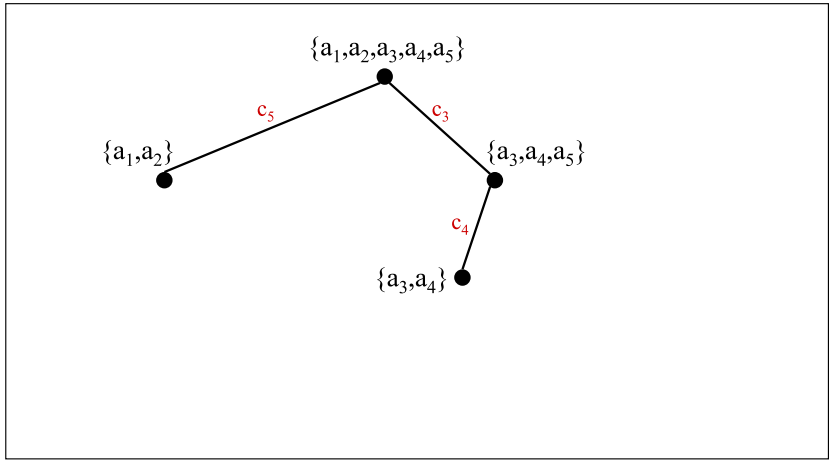
Construção de Filogenia Perfeita



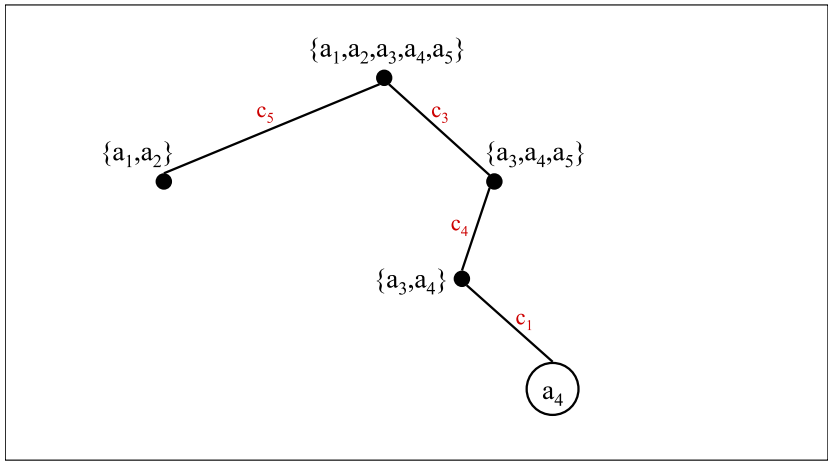
Construção de Filogenia Perfeita



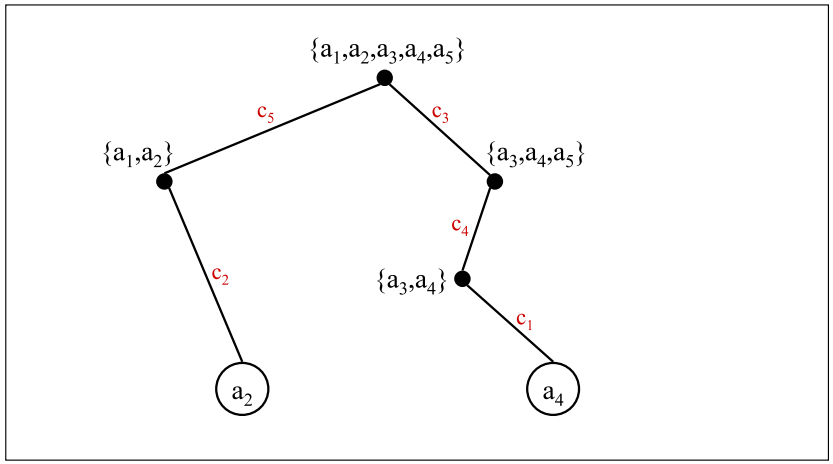
Construção de Filogenia Perfeita



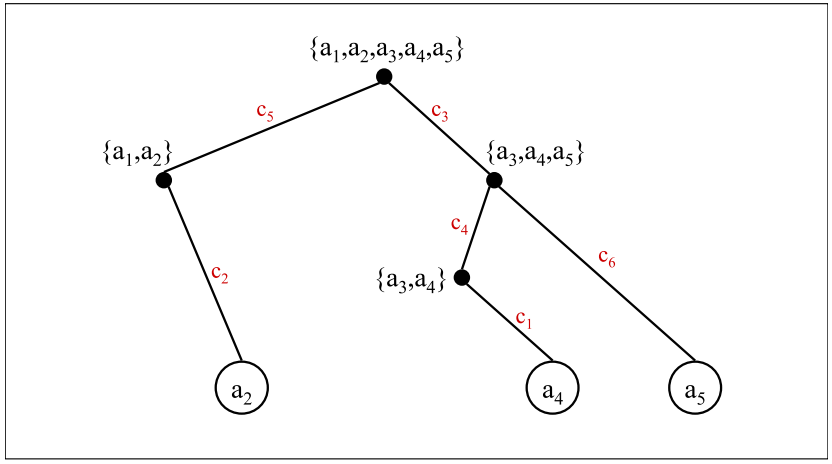
Construção de Filogenia Perfeita



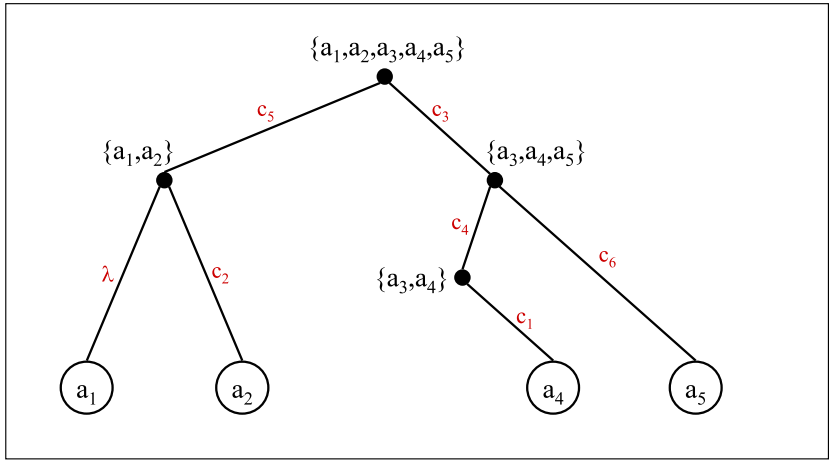
Construção de Filogenia Perfeita



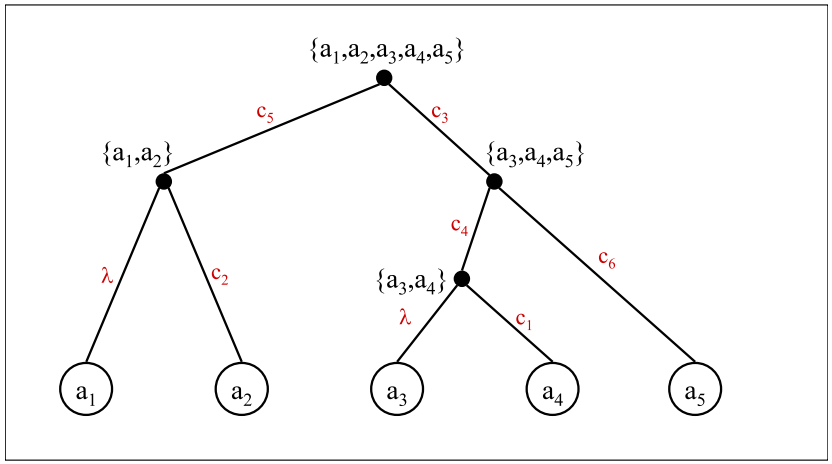
Construção de Filogenia Perfeita



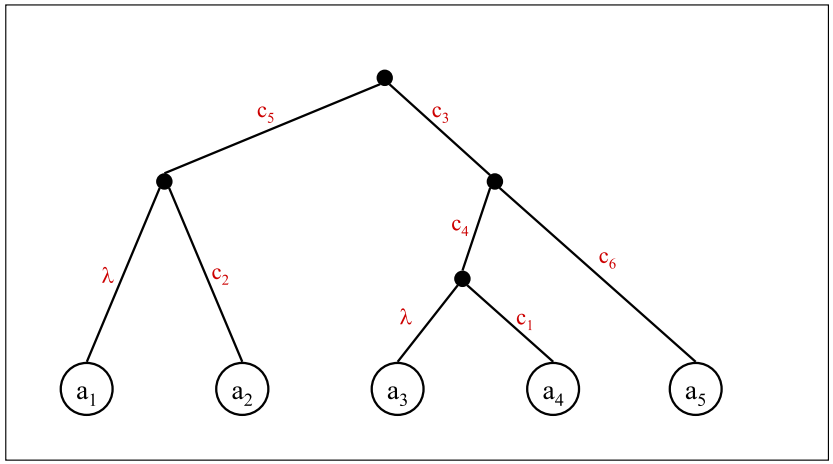
Construção de Filogenia Perfeita



Construção de Filogenia Perfeita



Construção de Filogenia Perfeita



Construção de Filogenia Perfeita

- Complexidade:
  - Ordenação da matriz de características binárias, segundo o número de 1's em cada coluna: $O(nm)$ (para contar o número de 1's em cada coluna) + $O(n+m)$ (para ordenar, usando Counting Sort) = $O(nm)$.
  - Busca do vértice X que representa o menor conjunto que contém $\mathcal{A}_j$ (para $1 \leq j \leq m$): $\sum_{j=1}^{m} O(nm) = O(nm^2)$.
  - Busca do vértice X que contém o objeto a_i (para $1 \leq i \leq n$), usando uma vetor auxiliar para armazenar o menor conjunto que contém cada objeto de $\mathcal{A}$: $O(nm)$ (para criar a tabela, a cada nova inserção de um vértice na árvore) + $n \times O(1) = O(n)$ (para acessar a tabela e criar as folhas faltantes) = $O(nm)$.
  - Total: = $O(nm^2)$.
- Algoritmo proposto por Dan Gusfield, em 1991. Neste mesmo trabalho, Gusfield apresenta um algoritmo capaz de testar se uma matriz binária admite filogenia perfeita em $O(nm)$.

PHYLIP

- *PHYLIP*: PHYLogeny Inference Package.
- Pacote gratuito e multiplataforma de análise filogenética desenvolvido Joseph Felsenstein em 1989, e mantida pela Universidade de Washington.
- É capaz de resolver a maioria das análises filogenéticas existentes na literatura atual.
- Aceita uma grande variedade de tipos de dados de entrada, como, por exemplo, sequências moleculares, frequência de genes, matriz de distância e características discretas.