

Anexos do Livro

Definições auxiliares

Professor Murilo V. G. da Silva

Departamento de Informática
Universidade Federal do Paraná

2025 / 2

Definições auxiliares para grafos

Definições auxiliares para grafos

Listas e Permutações:

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ...

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Também usamos a notação $A_1 = 1$, $A_2 = 5$, $A_3 = 9$, ...

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Também usamos a notação $A_1 = 1$, $A_2 = 5$, $A_3 = 9$, ...

- Listas, vetores e tuplas são tratadas como equivalentes

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Também usamos a notação $A_1 = 1$, $A_2 = 5$, $A_3 = 9$, ...

- Listas, vetores e tuplas são tratadas como equivalentes (strings também tratadas como tuplas quando conveniente)

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Também usamos a notação $A_1 = 1$, $A_2 = 5$, $A_3 = 9$, ...

- Listas, vetores e tuplas são tratadas como equivalentes (strings também tratadas como tuplas quando conveniente)
- Permutação de um conjunto S com n elementos:

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Também usamos a notação $A_1 = 1$, $A_2 = 5$, $A_3 = 9$, ...

- Listas, vetores e tuplas são tratadas como equivalentes (strings também tratadas como tuplas quando conveniente)
- Permutação de um conjunto S com n elementos:
Definido como uma **lista de n elementos de S sem repetição**.

Definições auxiliares para grafos

Listas e Permutações:

- Para listas, usamos colchetes

Exemplo: $A = [1, 5, 9, 7, 1]$ (existe ordem e possibilidade de repetição)

Usamos a notação $A[1] = 1$, $A[2] = 5$, $A[3] = 9$, ... (semelhante a vetor)

Também usamos a notação $A_1 = 1$, $A_2 = 5$, $A_3 = 9$, ...

- Listas, vetores e tuplas são tratadas como equivalentes (strings também tratadas como tuplas quando conveniente)
- Permutação de um conjunto S com n elementos:
Definido como uma **lista de n elementos de S sem repetição**.

Grafos, grafos direcionados e grafos ponderados (anexo)

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

e.g.,

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

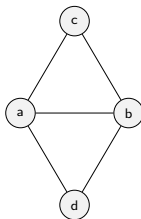
e.g., $G = (V(G), E(G))$, sendo $V(G) = \{a, b, c, d\}$ e $E(G) = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}\}$

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

e.g., $G=(V(G), E(G))$, sendo $V(G)=\{a, b, c, d\}$ e $E(G)=\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}\}$



Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de *conjunto de arestas*

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de pares de vértices, chamado de **conjunto de arcos**

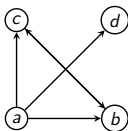
Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de **pares** de vértices, chamado de **conjunto de arcos**



$$V(G) = \{a, b, c, d\} \quad E(G) = \{(a, b), (a, c), (a, d), (b, c), (c, b)\}$$

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de *conjunto de arestas*

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de *pares* de vértices, chamado de *conjunto de arcos*

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de *conjunto de arestas*

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de *pares* de vértices, chamado de *conjunto de arcos*

Notação simplificada: $G = (V, E)$

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de **pares** de vértices, chamado de **conjunto de arcos**

Notação simplificada: $G = (V, E)$ ao invés de $G = (V(G), E(G))$

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de *conjunto de arestas*

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de *pares* de vértices, chamado de *conjunto de arcos*

Notação simplificada: $G = (V, E)$ ao invés de $G = (V(G), E(G))$

Quando o $V(G)$ não é definido explicitamente, a convenção é

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de **conjunto de vértices**
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de **conjunto de arestas**

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de **pares** de vértices, chamado de **conjunto de arcos**

Notação simplificada: $G = (V, E)$ ao invés de $G = (V(G), E(G))$

Quando o $V(G)$ não é definido explicitamente, a convenção é $V(G) = \{1, \dots, n\}$.

Grafos, grafos direcionados e grafos ponderados (anexo)

Um *grafo* G é um par de conjuntos $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de *conjunto de vértices*
- $E(G)$ é um conjunto onde cada elemento é um conjunto de dois vértices, chamado de *conjunto de arestas*

Um *grafo direcionado* G é um par $(V(G), E(G))$ onde

- $V(G)$ é um conjunto finito, chamado de conjunto de vértices
- $E(G)$ é um conjunto de *pares* de vértices, chamado de *conjunto de arcos*

Notação simplificada: $G = (V, E)$ ao invés de $G = (V(G), E(G))$

Quando o $V(G)$ não é definido explicitamente, a convenção é $V(G) = \{1, \dots, n\}$.

Notação simplificada: uv denota a aresta $\{u, v\}$

Grafos, grafos direcionados e grafos ponderados

Grafos, grafos direcionados e grafos ponderados

Um *grafo ponderado* é um par (G, w) onde G é um grafo e w é uma função que associa a cada aresta a de G um *peso* $w(a)$.

Grafos, grafos direcionados e grafos ponderados

Um *grafo ponderado* é um par (G, w) onde G é um grafo e w é uma função que associa a cada aresta a de G um *peso* $w(a)$.

- *Grafos direcionados ponderados* são definidos de maneira análoga.

Grafos, grafos direcionados e grafos ponderados

Um *grafo ponderado* é um par (G, w) onde G é um grafo e w é uma função que associa a cada aresta a de G um *peso* $w(a)$.

- **Grafos direcionados ponderados** são definidos de maneira análoga.
- Simplificando: às vezes diremos “**grafo ponderado G** ” ao invés de “grafo ponderado (G, w) ”.

Grafos, grafos direcionados e grafos ponderados

Um *grafo ponderado* é um par (G, w) onde G é um grafo e w é uma função que associa a cada aresta a de G um *peso* $w(a)$.

- **Grafos direcionados ponderados** são definidos de maneira análoga.
- Simplificando: às vezes diremos “**grafo ponderado G** ” ao invés de “grafo ponderado (G, w) ”.
- Convenção: Se G não é um grafo ponderado, $\forall e \in E(G)$, $w(e) = 1$.

Grafos, grafos direcionados e grafos ponderados

Um *grafo ponderado* é um par (G, w) onde G é um grafo e w é uma função que associa a cada aresta a de G um *peso* $w(a)$.

- **Grafos direcionados ponderados** são definidos de maneira análoga.
- Simplificando: às vezes diremos “**grafo ponderado G** ” ao invés de “grafo ponderado (G, w) ”.
- Convenção: Se G não é um grafo ponderado, $\forall e \in E(G)$, $w(e) = 1$.
- Notação simplificada: Se $\{u, v\} \in E(G)$, escrevemos $w(u, v)$ para o peso de $\{u, v\}$.

Grafos, grafos direcionados e grafos ponderados

Um *grafo ponderado* é um par (G, w) onde G é um grafo e w é uma função que associa a cada aresta a de G um *peso* $w(a)$.

- **Grafos direcionados ponderados** são definidos de maneira análoga.
- Simplificando: às vezes diremos “**grafo ponderado G** ” ao invés de “grafo ponderado (G, w) ”.
- Convenção: Se G não é um grafo ponderado, $\forall e \in E(G)$, $w(e) = 1$.
- Notação simplificada: Se $\{u, v\} \in E(G)$, escrevemos $w(u, v)$ para o peso de $\{u, v\}$. (ao invés de escrever $w(\{u, v\})$)

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G :

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

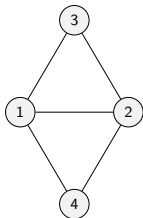
Exemplo:

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Exemplo: Seja G o grafo

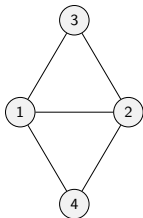


Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Exemplo: Seja G o grafo



$$M_G = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G :

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G : um conjunto $S \subseteq V(G)$ tal que
 $\forall u, v \in S, uv \in E(G)$

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G : um conjunto $S \subseteq V(G)$ tal que
 $\forall u, v \in S, uv \in E(G)$ (i.e., todo par de vértices em S é adjacente)

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G : um conjunto $S \subseteq V(G)$ tal que $\forall u, v \in S, uv \in E(G)$ (i.e., todo par de vértices em S é adjacente)

Um conjunto independente em um grafo G :

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G : um conjunto $S \subseteq V(G)$ tal que
 $\forall u, v \in S, uv \in E(G)$ (i.e., todo par de vértices em S é adjacente)

Um conjunto independente em um grafo G : um conjunto $S \subseteq V(G)$ tal que $\forall u, v \in S, uv \notin E(G)$

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G : um conjunto $S \subseteq V(G)$ tal que $\forall u, v \in S, uv \in E(G)$ (i.e., todo par de vértices em S é adjacente)

Um conjunto independente em um grafo G : um conjunto $S \subseteq V(G)$ tal que $\forall u, v \in S, uv \notin E(G)$
(i.e., nenhum par de vértices em S é adjacente)

Matriz de Adjacência, propriedades de grafos

Matriz de Adjacência de um grafo G : A *matriz de adjacência* de G é a matriz M_G indexada por $V(G) \times V(G)$ dada por

$$M_G[u, v] = \begin{cases} w(u, v), & uv \in E(G), \\ 0, & uv \notin E(G). \end{cases}$$

Note: A definição de matriz de adjacência para grafos ponderados é a mesmo

Uma clique em um grafo G : um conjunto $S \subseteq V(G)$ tal que $\forall u, v \in S, uv \in E(G)$ (i.e., todo par de vértices em S é adjacente)

Um conjunto independente em um grafo G : um conjunto $S \subseteq V(G)$ tal que $\forall u, v \in S, uv \notin E(G)$
(i.e., nenhum par de vértices em S é adjacente)

Matriz de Adjacência, propriedades de grafos

Grafo completo:

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo:

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano:

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)
(definição precisa no próximo slide)

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)
(definição precisa no próximo slide)

Grafo euleriano:

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)
(definição precisa no próximo slide)

Grafo euleriano: existe passeio visitando todas as arestas (sem repetir)

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)
(definição precisa no próximo slide)

Grafo euleriano: existe passeio visitando todas as arestas (sem repetir)
(definição precisa no próximo slide)

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)
(definição precisa no próximo slide)

Grafo euleriano: existe passeio visitando todas as arestas (sem repetir)
(definição precisa no próximo slide)

Árvore:

Matriz de Adjacência, propriedades de grafos

Grafo completo: existe uma aresta entre cada par de vértices do grafo
(i.e., $V(G)$ é uma clique)

Grafo conexo: existe um caminho entre cada par de vértices

Grafo hamiltoniano: existe ciclo visitando todos os vértices (sem repetir)
(definição precisa no próximo slide)

Grafo euleriano: existe passeio visitando todas as arestas (sem repetir)
(definição precisa no próximo slide)

Árvore: grafo conexo e acíclico

Grafos Hamiltonianos

Grafos Hamiltonianos

Seja G um grafo com n vértices

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano:

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

- Isto é, existe um ciclo passando por todos os vértices do grafo

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

- Isto é, existe um ciclo passando por todos os vértices do grafo
- Grafos que admitem circuitos hamiltonianos são chamados de **grafos hamiltonianos**.

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

- Isto é, existe um ciclo passando por todos os vértices do grafo
- Grafos que admitem circuitos hamiltonianos são chamados de **grafos hamiltonianos**.

Definição de Caminho Hamiltoniano:

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

- Isto é, existe um ciclo passando por todos os vértices do grafo
- Grafos que admitem circuitos hamiltonianos são chamados de **grafos hamiltonianos**.

Definição de Caminho Hamiltoniano: A permutação π é um **caminho hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n - 1$.

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

- Isto é, existe um ciclo passando por todos os vértices do grafo
- Grafos que admitem circuitos hamiltonianos são chamados de **grafos hamiltonianos**.

Definição de Caminho Hamiltoniano: A permutação π é um **caminho hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n - 1$.

- Isto é, existe um caminho passando por todos os vértices do grafo

Grafos Hamiltonianos

Seja G um grafo com n vértices

Seja $\pi = [\pi_1, \dots, \pi_n]$ uma permutação de $V(G)$.

- obs: o elemento π_{n+1} se refere à π_1 .

Definição de Circuito Hamiltoniano: A permutação π é um **circuito hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n$.

- Isto é, existe um ciclo passando por todos os vértices do grafo
- Grafos que admitem circuitos hamiltonianos são chamados de **grafos hamiltonianos**.

Definição de Caminho Hamiltoniano: A permutação π é um **caminho hamiltonino** de G se $\{\pi_i, \pi_{i+1}\} \in E(G)$, $i = 1, \dots, n - 1$.

- Isto é, existe um caminho passando por todos os vértices do grafo (não necessariamente voltando ao vértice inicial)

Grafos Eulerianos

Grafos Eulerianos

Seja G um grafo com m arestas

Grafos Eulerianos

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano:

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano: A permutação π é um **circuito euleriano** de G

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano: A permutação π é um **circuito euleriano** de G se para todo $i \in [1..n]$, $\pi_i = uv$ e $\pi_{i+1} = vw$,

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano: A permutação π é um **circuito euleriano** de G se para todo $i \in [1..n]$, $\pi_i = uv$ e $\pi_{i+1} = vw$, $u, v, w \in V(G)$.

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano: A permutação π é um **circuito euleriano** de G se para todo $i \in [1..n]$, $\pi_i = uv$ e $\pi_{i+1} = vw$, $u, v, w \in V(G)$.

- Isto é, existe um passeio visitando todas as arestas do grafo

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano: A permutação π é um **circuito euleriano** de G se para todo $i \in [1..n]$, $\pi_i = uv$ e $\pi_{i+1} = vw$, $u, v, w \in V(G)$.

- Isto é, existe um passeio visitando todas as arestas do grafo
- Grafos que admitem circuitos eulerianos são chamados de **grafos eulerianos**.

Seja G um grafo com m arestas

Seja $\pi = [\pi_1, \dots, \pi_m]$ uma permutação de $E(G)$.

- obs: o elemento π_{m+1} se refere à π_1 .

Definição de Circuito Euleriano: A permutação π é um **circuito euleriano** de G se para todo $i \in [1..n]$, $\pi_i = uv$ e $\pi_{i+1} = vw$, $u, v, w \in V(G)$.

- Isto é, existe um passeio visitando todas as arestas do grafo
- Grafos que admitem circuitos eulerianos são chamados de **grafos eulerianos**.

Problema de Decisão em Grafos

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Para problemas de decisão em geral (não apenas sobre grafos):

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Para problemas de decisão em geral (não apenas sobre grafos):

x é uma instância verdadeira do problema L :

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Para problemas de decisão em geral (não apenas sobre grafos):

x é uma instância verdadeira do problema L : significa que $x \in L$.

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Para problemas de decisão em geral (não apenas sobre grafos):

x é uma instância verdadeira do problema L : significa que $x \in L$.

x é uma instância falsa do problema L :

Problema de Decisão em Grafos

Exemplos de problemas de decisão em grafos:

- $L_{\text{CNX}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo conexo} \}.$
- $L_{\text{EU}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo euleriano} \}$
- $L_{\text{HAM}} = \{ \langle G \rangle \in \Sigma^* : G \text{ é um grafo hamiltoniano} \}.$

Para problemas de decisão em geral (não apenas sobre grafos):

x é uma instância verdadeira do problema L : significa que $x \in L$.

x é uma instância falsa do problema L : $x \notin L$.

Responda as questões abaixo:

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$ ▶

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$ ▶

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$ ▶

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$ ▶

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore?
- 4 Desenhe todos os grafos de 4 vértices¹.

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore?
- 4 Desenhe todos os grafos de 4 vértices¹.
- 5 Apresente um grafo de 4 vértices que seja conexo, mas que não seja nem árvore nem grafo hamiltoniano.

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore?
- 4 Desenhe todos os grafos de 4 vértices¹.
- 5 Apresente um grafo de 4 vértices que seja conexo, mas que não seja nem árvore nem grafo hamiltoniano.
- 6 Enumere todas as cliques do grafo da resposta da Questão 5.

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore?
- 4 Desenhe todos os grafos de 4 vértices¹.
- 5 Apresente um grafo de 4 vértices que seja conexo, mas que não seja nem árvore nem grafo hamiltoniano.
- 6 Enumere todas as cliques do grafo da resposta da Questão 5.
- 7 É verdade que todo grafo completo também é um grafo hamiltoniano?

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore?
- 4 Desenhe todos os grafos de 4 vértices¹.
- 5 Apresente um grafo de 4 vértices que seja conexo, mas que não seja nem árvore nem grafo hamiltoniano.
- 6 Enumere todas as cliques do grafo da resposta da Questão 5.
- 7 É verdade que todo grafo completo também é um grafo hamiltoniano?
- 8 É verdade que todo grafo com pelo menos 6 vértices tem uma clique de tamanho 3 ou um conjunto independente de tamanho 3?

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$

Responda as questões abaixo:

- 1 Quantas arestas contém um grafo completo com n vértices?
- 2 É verdade que todo grafo completo também é conexo?
- 3 É verdade que todo grafo conexo também é uma árvore?
- 4 Desenhe todos os grafos de 4 vértices¹.
- 5 Apresente um grafo de 4 vértices que seja conexo, mas que não seja nem árvore nem grafo hamiltoniano.
- 6 Enumere todas as cliques do grafo da resposta da Questão 5.
- 7 É verdade que todo grafo completo também é um grafo hamiltoniano?
- 8 É verdade que todo grafo com pelo menos 6 vértices tem uma clique de tamanho 3 ou um conjunto independente de tamanho 3 (ou ambos).

¹ Lembre: quando conjunto de vértices do grafo não é especificado, por convenção usamos naturais $1, 2, 3, \dots$

Problema SAT

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$,

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “fórmulas CNF”

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “fórmulas CNF”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “fórmulas CNF”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “fórmulas CNF”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ .

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$,

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V.

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V.

Note: ϕ_1 é satisfazível:

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V .

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V .

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

$$L_{\text{SAT}} = \{\perp\phi \in \Sigma^* : \phi \text{ é uma fórmula CNF satisfazível}\}.$$

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V .

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

$$L_{\text{SAT}} = \{\ulcorner \phi \urcorner \in \Sigma^* : \phi \text{ é uma fórmula CNF satisfazível}\}.$$

Como a fórmula ϕ_1 é satisfazível, a string $\ulcorner \phi_1 \urcorner$ é uma instância verdadeira de L_{SAT} .

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V .

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

$$L_{\text{SAT}} = \{\ulcorner \phi \urcorner \in \Sigma^* : \phi \text{ é uma fórmula CNF satisfazível}\}.$$

Como a fórmula ϕ_1 é satisfazível, a string $\ulcorner \phi_1 \urcorner$ é uma instância verdadeira de L_{SAT} .

Seja $\phi_2 = (\overline{x_1} \vee \overline{x_2}) \wedge (x_1) \wedge (x_2)$.

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “fórmulas CNF”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V.

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

$$L_{\text{SAT}} = \{\ulcorner \phi \urcorner \in \Sigma^* : \phi \text{ é uma fórmula CNF satisfazível}\}.$$

Como a fórmula ϕ_1 é satisfazível, a string $\ulcorner \phi_1 \urcorner$ é uma instância verdadeira de L_{SAT} .

Seja $\phi_2 = (\overline{x_1} \vee \overline{x_2}) \wedge (x_1) \wedge (x_2)$. Note: $\ulcorner \phi_2 \urcorner$ é uma instância falsa de L_{SAT} .

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V .

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

$$L_{\text{SAT}} = \{\ulcorner \phi \urcorner \in \Sigma^* : \phi \text{ é uma fórmula CNF satisfazível}\}.$$

Como a fórmula ϕ_1 é satisfazível, a string $\ulcorner \phi_1 \urcorner$ é uma instância verdadeira de L_{SAT} .

Seja $\phi_2 = (\overline{x_1} \vee \overline{x_2}) \wedge (x_1) \wedge (x_2)$. Note: $\ulcorner \phi_2 \urcorner$ é uma instância falsa de L_{SAT} .

- Terminologia:** Cada disjunção em uma fórmula CNF é chamada de *cláusula*

Problema SAT

Dadas variáveis booleanas $x_1, \dots, x_n$, uma *fórmula booleana na forma normal conjuntiva* é uma conjunção de disjunções de termos que podem ser x_i ou $\overline{x_i}$.

- Exemplo: $\phi_1 = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_3})$
- Para simplificar, diremos “*fórmulas CNF*”
- Os termos x_i ou $\overline{x_i}$ são chamados de *literais* da fórmula.
e.g. os literais de ϕ_1 são $x_1, x_2, \overline{x_2}$ e $\overline{x_3}$

Satisfatibilidade de fórmulas booleanas (SAT): O problema de decidir se existe uma valoração que satisfaz uma dada fórmula CNF ϕ . Isto é, $\exists v_1, v_2, \dots, v_n$, sendo $v_i \in \{V, F\}$, de forma que substituindo v_i na variável x_i , o valor de ϕ é V .

Note: ϕ_1 é satisfazível: tome $v_1 = V, v_2 = V, v_3 = F$

$$L_{\text{SAT}} = \{\ulcorner \phi \urcorner \in \Sigma^* : \phi \text{ é uma fórmula CNF satisfazível}\}.$$

Como a fórmula ϕ_1 é satisfazível, a string $\ulcorner \phi_1 \urcorner$ é uma instância verdadeira de L_{SAT} .

Seja $\phi_2 = (\overline{x_1} \vee \overline{x_2}) \wedge (x_1) \wedge (x_2)$. Note: $\ulcorner \phi_2 \urcorner$ é uma instância falsa de L_{SAT} .

- Terminologia:** Cada disjunção em uma fórmula CNF é chamada de *cláusula*
e.g. as cláusulas de ϕ_2 são $(\overline{x_1} \vee \overline{x_2})$, (x_1) e (x_2)