

Algorytmy i struktury danych - Grafy

Marcin Żurowski

24 kwietnia 2025

Plan zajęć

- 1 Graf
- 2 Reprezentacja grafu
- 3 Własności grafu
- 4 Drzewa
- 5 Reprezentacja drzewa binarnego
- 6 Drzewa uporządkowane
- 7 Drzewa decyzyjne

Graf

Graf (nieskierowany) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych podzbiorów zbioru V .

Krawędź grafu jest zbiorem $\{u, v\}$, gdzie $u, v \in V$

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako linie między punktami.

Graf

Graf (nieskierowany) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podzbiorów** zbioru V .

Krawędź grafu jest zbiorem $\{u, v\}$, gdzie $u, v \in V$

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **linie** między punktami.

Graf

Graf (nieskierowany) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podzbiorów** zbioru V .

Krawędź grafu jest zbiorem $\{u, v\}$, gdzie $u, v \in V$

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **linie** między punktami.

Graf

Graf (nieskierowany) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podzbiorów** zbioru V .

Krawędź grafu jest zbiorem $\{u, v\}$, gdzie $u, v \in V$

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **linie** między punktami.

Graf

Graf (nieskierowany) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podzbiorów** zbioru V .

Krawędź grafu jest zbiorem $\{u, v\}$, gdzie $u, v \in V$

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **linie** między punktami.

Graf

Graf skierowany (digraf) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych podciągów zbioru V .

Krawędź grafu jest ciągiem (u, v) , gdzie $u, v \in V$. Zatem (u, v) i (v, u) to dwie różne krawędzie.

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako strzałki między punktami.

Graf

Graf skierowany (digraf) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podciągów** zbioru V .

Krawędź grafu jest ciągiem (u, v) , gdzie $u, v \in V$. Zatem (u, v) i (v, u) to dwie różne krawędzie.

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **strzałki** między punktami.

Graf

Graf skierowany (digraf) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podciągów** zbioru V .

Krawędź grafu jest ciągiem (u, v) , gdzie $u, v \in V$. Zatem (u, v) i (v, u) to dwie różne krawędzie.

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **strzałki** między punktami.

Graf

Graf skierowany (digraf) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podciągów** zbioru V .

Krawędź grafu jest ciągiem (u, v) , gdzie $u, v \in V$. Zatem (u, v) i (v, u) to dwie różne krawędzie.

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **strzałki** między punktami.

Graf

Graf skierowany (digraf) to para zbiorów $G = (V, E)$:

- V to skończony zbiór wierzchołków grafu G
- E to skończony zbiór krawędzi, czyli dwuelementowych **podciągów** zbioru V .

Krawędź grafu jest ciągiem (u, v) , gdzie $u, v \in V$. Zatem (u, v) i (v, u) to dwie różne krawędzie.

Graf możemy narysować. Wierzchołki rysujemy jako punkty, a krawędzie jako **strzałki** między punktami.

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

- wierzchołek v jest następnikiem wierzchołka u (a wierzchołek u jest poprzednikiem wierzchołka v)
- krawędź (u, v) wychodzi z wierzchołka u i wchodzi do wierzchołka v

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

- wierzchołek v jest następnikiem wierzchołka u (a wierzchołek u jest poprzednikiem wierzchołka v)
- krawędź (u, v) wychodzi z wierzchołka u i wchodzi do wierzchołka v

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

- wierzchołek v jest następnikiem wierzchołka u (a wierzchołek u jest poprzednikiem wierzchołka v)
- krawędź (u, v) wychodzi z wierzchołka u i wchodzi do wierzchołka v

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

- wierzchołek v jest następnikiem wierzchołka u (a wierzchołek u jest poprzednikiem wierzchołka v)
- krawędź (u, v) wychodzi z wierzchołka u i wchodzi do wierzchołka v

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

- wierzchołek v jest następnikiem wierzchołka u (a wierzchołek u jest poprzednikiem wierzchołka v)
- krawędź (u, v) wychodzi z wierzchołka u i wchodzi do wierzchołka v

Graf

Jeśli u, v są wierzchołkami grafu nieskierowanego G i $\{u, v\} \in E$, to mówimy, że:

- wierzchołki u i v są sąsiadami
- krawędź $\{u, v\}$ łączy wierzchołki u i v

Jeśli u, v są wierzchołkami grafu skierowanego G i $(u, v) \in E$, to mówimy, że:

- wierzchołek v jest następnikiem wierzchołka u (a wierzchołek u jest poprzednikiem wierzchołka v)
- krawędź (u, v) wychodzi z wierzchołka u i wchodzi do wierzchołka v

Reprezentacja grafu

Macierz sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$.
Przedstawiamy graf G za pomocą macierzy A rozmiaru $|V| \times |V|$, takie, że $A[u, v] = 1$ jeśli $(u, v) \in E$, i $A[u, v] = 0$ jeśli $(u, v) \notin E$.
- Zaleta: można sprawdzić czy istnieje krawędź w grafie w czasie $\Theta(1)$
- Wady:

Reprezentacja grafu

Macierz sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$.
Przedstawiamy graf G za pomocą macierzy A rozmiaru $|V| \times |V|$, takie, że $A[u, v] = 1$ jeśli $(u, v) \in E$, i $A[u, v] = 0$ jeśli $(u, v) \notin E$.
- Zaleta: można sprawdzić czy istnieje krawędź w grafie w czasie $\Theta(1)$
- Wady:
 - zużycie $\Theta(|V|^2)$ pamięci
 - odnalezienie wszystkich sąsiadów wierzchołka v zajmuje czas $\Theta(|V|)$ nawet jeśli tych sąsiadów jest znacznie mniej niż $|V|$.

Reprezentacja grafu

Macierz sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$.
Przedstawiamy graf G za pomocą macierzy A rozmiaru $|V| \times |V|$, takie, że $A[u, v] = 1$ jeśli $(u, v) \in E$, i $A[u, v] = 0$ jeśli $(u, v) \notin E$.
- Zaleta: można sprawdzić czy istnieje krawędź w grafie w czasie $\Theta(1)$
- Wady:
 - zużycie $\Theta(|V|^2)$ pamięci
 - odnalezienie wszystkich sąsiadów wierzchołka v zajmuje czas $\Theta(|V|)$ nawet jeśli tych sąsiadów jest znacznie mniej niż $|V|$.

Reprezentacja grafu

Macierz sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$.
Przedstawiamy graf G za pomocą macierzy A rozmiaru $|V| \times |V|$, takie, że $A[u, v] = 1$ jeśli $(u, v) \in E$, i $A[u, v] = 0$ jeśli $(u, v) \notin E$.
- Zaleta: można sprawdzić czy istnieje krawędź w grafie w czasie $\Theta(1)$
- Wady:
 - zużycie $\Theta(|V|^2)$ pamięci
 - odnalezienie wszystkich sąsiadów wierzchołka v zajmuje czas $\Theta(|V|)$ nawet jeśli tych sąsiadów jest znacznie mniej niż $|V|$.

Reprezentacja grafu

Macierz sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$.
Przedstawiamy graf G za pomocą macierzy A rozmiaru $|V| \times |V|$, takie, że $A[u, v] = 1$ jeśli $(u, v) \in E$, i $A[u, v] = 0$ jeśli $(u, v) \notin E$.
- Zaleta: można sprawdzić czy istnieje krawędź w grafie w czasie $\Theta(1)$
- Wady:
 - zużycie $\Theta(|V|^2)$ pamięci
 - odnalezienie wszystkich sąsiadów wierzchołka v zajmuje czas $\Theta(|V|)$ nawet jeśli tych sąsiadów jest znacznie mniej niż $|V|$.

Reprezentacja grafu

Lista sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$. Przedstawiamy graf G za pomocą tablicy A rozmiaru $|V|$, zawierającej wskaźniki do $|V|$ list. Lista $A[u]$ zawiera wszystkie wierzchołki v_i , dla których $(u, v_i) \in E$.
- Zaleta:
 - zużywa tylko $O(|V| + |E|)$ pamięci
 - łatwa iteracja po wszystkich sąsiadach (następnikach) zadanego wierzchołka v
- Wady: sprawdzenie, czy krawędź (u, v) występuje w grafie, wymaga w najgorszym przypadku przejrzania całej listy $A[u]$.

Reprezentacja grafu

Lista sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$. Przedstawiamy graf G za pomocą tablicy A rozmiaru $|V|$, zawierającej wskaźniki do $|V|$ list. Lista $A[u]$ zawiera wszystkie wierzchołki v_i , dla których $(u, v_i) \in E$.
- Zaleta:
 - zużywa tylko $O(|V| + |E|)$ pamięci
 - łatwa iteracja po wszystkich sąsiadach (następnikach) zadanego wierzchołka v
- Wady: sprawdzenie, czy krawędź (u, v) występuje w grafie, wymaga w najgorszym przypadku przejrzenia całej listy $A[u]$.

Reprezentacja grafu

Lista sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$. Przedstawiamy graf G za pomocą tablicy A rozmiaru $|V|$, zawierającej wskaźniki do $|V|$ list. Lista $A[u]$ zawiera wszystkie wierzchołki v_i , dla których $(u, v_i) \in E$.
- Zaleta:
 - zużywa tylko $O(|V| + |E|)$ pamięci
 - łatwa iteracja po wszystkich sąsiadach (następnikach) zadanego wierzchołka v
- Wady: sprawdzenie, czy krawędź (u, v) występuje w grafie, wymaga w najgorszym przypadku przejrzania całej listy $A[u]$.

Reprezentacja grafu

Lista sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$. Przedstawiamy graf G za pomocą tablicy A rozmiaru $|V|$, zawierającej wskaźniki do $|V|$ list. Lista $A[u]$ zawiera wszystkie wierzchołki v_i , dla których $(u, v_i) \in E$.
- Zaleta:
 - zużywa tylko $O(|V| + |E|)$ pamięci
 - łatwa iteracja po wszystkich sąsiadach (następnikach) zadanego wierzchołka v
- Wady: sprawdzenie, czy krawędź (u, v) występuje w grafie, wymaga w najgorszym przypadku przejrzenia całej listy $A[u]$.

Reprezentacja grafu

Lista sąsiedztwa:

- Zakładamy, że wierzchołki grafu są ponumerowane od 1 do $|V|$. Przedstawiamy graf G za pomocą tablicy A rozmiaru $|V|$, zawierającej wskaźniki do $|V|$ list. Lista $A[u]$ zawiera wszystkie wierzchołki v_i , dla których $(u, v_i) \in E$.
- Zaleta:
 - zużywa tylko $O(|V| + |E|)$ pamięci
 - łatwa iteracja po wszystkich sąsiadach (następnikach) zadanego wierzchołka v
- Wady: sprawdzenie, czy krawędź (u, v) występuje w grafie, wymaga w najgorszym przypadku przejrzenia całej listy $A[u]$.

Własności grafu

Stopień wierzchołka v grafu nieskierowanego G nazywamy liczbę krawędzi incydentnych z v .

Jeśli G jest grafem skierowanym, to dla wierzchołków v wyróżniamy:

- stopień wyjściowy - liczbę krawędzi wychodzących z v
- stopień wejściowy - liczbę krawędzi wchodzących do v
- stopień - suma stopni wyjściowych i wejściowych

Własności grafu

Stopień wierzchołka v grafu nieskierowanego G nazywamy liczbę krawędzi incydentnych z v .

Jeśli G jest grafem skierowanym, to dla wierzchołków v wyróżniamy:

- **stopień wyjściowy** - liczbę krawędzi wychodzących z v
- **stopień wejściowy** - liczbę krawędzi wchodzących do v
- **stopień** - suma stopni wyjściowych i wejściowych

Własności grafu

Stopień wierzchołka v grafu nieskierowanego G nazywamy liczbę krawędzi incydentnych z v .

Jeśli G jest grafem skierowanym, to dla wierzchołków v wyróżniamy:

- **stopień wyjściowy** - liczbę krawędzi wychodzących z v
- **stopień wejściowy** - liczbę krawędzi wchodzących do v
- **stopień** - suma stopni wyjściowych i wejściowych

Własności grafu

Stopień wierzchołka v grafu nieskierowanego G nazywamy liczbę krawędzi incydentnych z v .

Jeśli G jest grafem skierowanym, to dla wierzchołków v wyróżniamy:

- **stopień wyjściowy** - liczbę krawędzi wychodzących z v
- **stopień wejściowy** - liczbę krawędzi wchodzących do v
- **stopień** - suma stopni wyjściowych i wejściowych

Własności grafu

Ścieżka długości k z wierzchołka u do wierzchołka u' nazywamy ciąg wierzchołków $(v_0, v_1, \dots, v_k)$, taki że $u = v_0$, $u' = v_k$ oraz $(v_{i-1}, v_i) \in E$ dla $i = 1, 2, \dots, k$ i wszystkie te krawędzie są różne.

Jeśli istnieje ścieżka z wierzchołka u do wierzchołka u' , to mówimy, że wierzchołek u' jest osiągalny z u .

Ścieżkę nazywamy **ścieżką prostą**, jeśli wszystkie jej wierzchołki są różne.

Ścieżkę nazywamy **ścieżką zamkniętą**, jeśli $v_0 = v_k$

Ścieżkę zamkniętą zawierającą co najmniej jedną krawędź nazywamy **cyklem**.

Graf niezawierający cykli, nazywamy **grafem acyklicznym**.

Własności grafu

Ścieżka długości k z wierzchołka u do wierzchołka u' nazywamy ciąg wierzchołków $(v_0, v_1, \dots, v_k)$, taki że $u = v_0$, $u' = v_k$ oraz $(v_{i-1}, v_i) \in E$ dla $i = 1, 2, \dots, k$ i wszystkie te krawędzie są różne. Jeśli istnieje ścieżka z wierzchołka u do wierzchołka u' , to mówimy, że wierzchołek u' jest osiągalny z u .

Ścieżkę nazywamy **ścieżką prostą**, jeśli wszystkie jej wierzchołki są różne.

Ścieżkę nazywamy **ścieżką zamkniętą**, jeśli $v_0 = v_k$

Ścieżkę zamkniętą zawierającą co najmniej jedną krawędź nazywamy **cyklem**.

Graf niezawierający cykli, nazywamy **grafem acyklicznym**.

Własności grafu

Ścieżka długości k z wierzchołka u do wierzchołka u' nazywamy ciąg wierzchołków $(v_0, v_1, \dots, v_k)$, taki że $u = v_0$, $u' = v_k$ oraz $(v_{i-1}, v_i) \in E$ dla $i = 1, 2, \dots, k$ i wszystkie te krawędzie są różne. Jeśli istnieje ścieżka z wierzchołka u do wierzchołka u' , to mówimy, że wierzchołek u' jest osiągalny z u .

Ścieżkę nazywamy **ścieżką prostą**, jeśli wszystkie jej wierzchołki są różne.

Ścieżkę nazywamy **ścieżką zamkniętą**, jeśli $v_0 = v_k$

Ścieżkę zamkniętą zawierającą co najmniej jedną krawędź nazywamy **cyklem**.

Graf niezawierający cykli, nazywamy **grafem acyklicznym**.

Własności grafu

Ścieżka długości k z wierzchołka u do wierzchołka u' nazywamy ciąg wierzchołków $(v_0, v_1, \dots, v_k)$, taki że $u = v_0$, $u' = v_k$ oraz $(v_{i-1}, v_i) \in E$ dla $i = 1, 2, \dots, k$ i wszystkie te krawędzie są różne. Jeśli istnieje ścieżka z wierzchołka u do wierzchołka u' , to mówimy, że wierzchołek u' jest osiągalny z u .

Ścieżkę nazywamy **ścieżką prostą**, jeśli wszystkie jej wierzchołki są różne.

Ścieżkę nazywamy **ścieżką zamkniętą**, jeśli $v_0 = v_k$

Ścieżkę zamkniętą zawierającą co najmniej jedną krawędź nazywamy **cyklem**.

Graf niezawierający cykli, nazywamy **grafem acyklicznym**.

Własności grafu

Ścieżka długości k z wierzchołka u do wierzchołka u' nazywamy ciąg wierzchołków $(v_0, v_1, \dots, v_k)$, taki że $u = v_0$, $u' = v_k$ oraz $(v_{i-1}, v_i) \in E$ dla $i = 1, 2, \dots, k$ i wszystkie te krawędzie są różne. Jeśli istnieje ścieżka z wierzchołka u do wierzchołka u' , to mówimy, że wierzchołek u' jest osiągalny z u .

Ścieżkę nazywamy **ścieżką prostą**, jeśli wszystkie jej wierzchołki są różne.

Ścieżkę nazywamy **ścieżką zamkniętą**, jeśli $v_0 = v_k$

Ścieżkę zamkniętą zawierającą co najmniej jedną krawędź nazywamy **cyklem**.

Graf niezawierający cykli, nazywamy **grafem acyklicznym**.

Własności grafu

Ścieżka długości k z wierzchołka u do wierzchołka u' nazywamy ciąg wierzchołków $(v_0, v_1, \dots, v_k)$, taki że $u = v_0$, $u' = v_k$ oraz $(v_{i-1}, v_i) \in E$ dla $i = 1, 2, \dots, k$ i wszystkie te krawędzie są różne. Jeśli istnieje ścieżka z wierzchołka u do wierzchołka u' , to mówimy, że wierzchołek u' jest osiągalny z u .

Ścieżkę nazywamy **ścieżką prostą**, jeśli wszystkie jej wierzchołki są różne.

Ścieżkę nazywamy **ścieżką zamkniętą**, jeśli $v_0 = v_k$

Ścieżkę zamkniętą zawierającą co najmniej jedną krawędź nazywamy **cyklem**.

Graf niezawierający cykli, nazywamy **grafem acyklicznym**.

Własności grafu

Graf nieskierowany G nazywamy **spójnym**, jeśli każda para jego wierzchołków jest połączona ścieżką.

Składową spójności grafu G nazywamy maksymalny w sensie zawierania podgraf spójny tego grafu.

Acykliczny graf spójny nazywamy **drzewem**.

Graf acykliczny (zbiór drzew) nazywamy **lasem**.

Własności grafu

Graf nieskierowany G nazywamy **spójnym**, jeśli każda para jego wierzchołków jest połączona ścieżką.

Składową spójności grafu G nazywamy maksymalny w sensie zawierania podgraf spójny tego grafu.

Acykliczny graf spójny nazywamy **drzewem**.

Graf acykliczny (zbiór drzew) nazywamy **lasem**.

Własności grafu

Graf nieskierowany G nazywamy **spójnym**, jeśli każda para jego wierzchołków jest połączona ścieżką.

Składową spójności grafu G nazywamy maksymalny w sensie zawierania podgraf spójny tego grafu.

Acykliczny graf spójny nazywamy **drzewem**.

Graf acykliczny (zbiór drzew) nazywamy **lasem**.

Własności grafu

Graf nieskierowany G nazywamy **spójnym**, jeśli każda para jego wierzchołków jest połączona ścieżką.

Składową spójności grafu G nazywamy maksymalny w sensie zawierania podgraf spójny tego grafu.

Acykliczny graf spójny nazywamy **drzewem**.

Graf acykliczny (zbiór drzew) nazywamy **lasem**.

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Drzewo z korzeniem (drzewo ukorzenione) - drzewo, w którym jeden z wierzchołków jest wyróżniony jako korzeń drzewa. Wierzchołki drzew ukorzenionych będziemy nazywać węzłami.

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Drzewo z korzeniem (drzewo ukorzenione) - drzewo, w którym jeden z wierzchołków jest wyróżniony jako korzeń drzewa. Wierzchołki drzew ukorzenionych będziemy nazywać węzłami.

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Drzewo z korzeniem (drzewo ukorzenione) - drzewo, w którym jeden z wierzchołków jest wyróżniony jako korzeń drzewa. Wierzchołki drzew ukorzenionych będziemy nazywać węzłami.

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Drzewo z korzeniem (drzewo ukorzenione) - drzewo, w którym jeden z wierzchołków jest wyróżniony jako korzeń drzewa.

Wierzchołki drzew ukorzenionych będziemy nazywać węzłami.

Własności drzew

Drzewo - acykliczny graf spójny. Jeśli graf $G = (V, E)$ jest drzewem, to:

- każde dwa wierzchołki drzewa G są połączone ze sobą dokładnie jedną ścieżką prostą
- G jest spójny, i gdy usuniemy dowolną krawędź z E to powstaje graf niespójny
- G jest acykliczny, ale jeśli dodamy jakąkolwiek krawędź do E , to powstanie cykl
- $|E| = |V| - 1$

Drzewo z korzeniem (drzewo ukorzenione) - drzewo, w którym jeden z wierzchołków jest wyróżniony jako korzeń drzewa. Wierzchołki drzew ukorzenionych będziemy nazywać węzłami.

Własności drzew

Niech x będzie dowolnym węzłem drzewa ukorzonego T . Każdy węzeł y na ścieżce z korzenia r do x nazywamy przodkiem węzła x .

Jeśli y jest przodkiem x , to x jest potomkiem węzła y .

Każdy węzeł jest jednocześnie swoim przodkiem i potomkiem.

Jeśli y jest przodkiem x i $x \neq y$, to y jest właściwym przodkiem węzła x , a x jest właściwym potomkiem węzła y .

Poddrzewo o korzeniu x jest drzewem, utworzonym z potomków węzła x , którego korzeniem jest x .

Własności drzew

Niech x będzie dowolnym węzłem drzewa ukorzonego T . Każdy węzeł y na ścieżce z korzenia r do x nazywamy przodkiem węzła x . Jeśli y jest przodkiem x , to x jest potomkiem węzła y .

Każdy węzeł jest jednocześnie swoim przodkiem i potomkiem.

Jeśli y jest przodkiem x i $x \neq y$, to y jest właściwym przodkiem węzła x , a x jest właściwym potomkiem węzła y .

Poddrzewo o korzeniu x jest drzewem, utworzonym z potomków węzła x , którego korzeniem jest x .

Własności drzew

Niech x będzie dowolnym węzłem drzewa ukorzonego T . Każdy węzeł y na ścieżce z korzenia r do x nazywamy przodkiem węzła x . Jeśli y jest przodkiem x , to x jest potomkiem węzła y .

Każdy węzeł jest jednocześnie swoim przodkiem i potomkiem.

Jeśli y jest przodkiem x i $x \neq y$, to y jest właściwym przodkiem węzła x , a x jest właściwym potomkiem węzła y .

Poddrzewo o korzeniu x jest drzewem, utworzonym z potomków węzła x , którego korzeniem jest x .

Własności drzew

Niech x będzie dowolnym węzłem drzewa ukorzonego T . Każdy węzeł y na ścieżce z korzenia r do x nazywamy przodkiem węzła x . Jeśli y jest przodkiem x , to x jest potomkiem węzła y .

Każdy węzeł jest jednocześnie swoim przodkiem i potomkiem.

Jeśli y jest przodkiem x i $x \neq y$, to y jest właściwym przodkiem węzła x , a x jest właściwym potomkiem węzła y .

Poddrzewo o korzeniu x jest drzewem, utworzonym z potomków węzła x , którego korzeniem jest x .

Własności drzew

Niech x będzie dowolnym węzłem drzewa ukorzonego T . Każdy węzeł y na ścieżce z korzenia r do x nazywamy przodkiem węzła x . Jeśli y jest przodkiem x , to x jest potomkiem węzła y .

Każdy węzeł jest jednocześnie swoim przodkiem i potomkiem.

Jeśli y jest przodkiem x i $x \neq y$, to y jest właściwym przodkiem węzła x , a x jest właściwym potomkiem węzła y .

Poddrzewo o korzeniu x jest drzewem, utworzonym z potomków węzła x , którego korzeniem jest x .

Własności drzew

Jeśli ostatnią krawędzią drzewa T na ścieżce od korzenia r do wężła x jest (y, x) , to y jest ojcem (rodzicem) wężła x , a x jest synem (dzieckiem) wężła y .

Korzeń drzewa to jedyny wężel, który nie ma ojca.

Dwa wężły mające tego samego ojca nazywamy braćmi.

Wężel, który nie ma synów, nazywamy liściem lub wężlem zewnętrznym.

Wężel, który nie jest liściem, nazywamy wężlem wewnętrznym.

Własności drzew

Jeśli ostatnią krawędzią drzewa T na ścieżce od korzenia r do wężła x jest (y, x) , to y jest ojcem (rodzicem) wężła x , a x jest synem (dzieckiem) wężła y .

Korzeń drzewa to jedyny wężel, który nie ma ojca.

Dwa wężły mające tego samego ojca nazywamy braćmi.

Wężel, który nie ma synów, nazywamy liściem lub wężlem zewnętrznym.

Wężel, który nie jest liściem, nazywamy wężlem wewnętrznym.

Własności drzew

Jeśli ostatnią krawędzią drzewa T na ścieżce od korzenia r do wężła x jest (y, x) , to y jest ojcem (rodzicem) wężła x , a x jest synem (dzieckiem) wężła y .

Korzeń drzewa to jedyny wężel, który nie ma ojca.

Dwa wężły mające tego samego ojca nazywamy braćmi.

Wężel, który nie ma synów, nazywamy liściem lub wężłem zewnętrznym.

Wężel, który nie jest liściem, nazywamy wężłem wewnętrznym.

Własności drzew

Jeśli ostatnią krawędzią drzewa T na ścieżce od korzenia r do wężła x jest (y, x) , to y jest ojcem (rodzicem) wężła x , a x jest synem (dzieckiem) wężła y .

Korzeń drzewa to jedyny wężel, który nie ma ojca.

Dwa wężły mające tego samego ojca nazywamy braćmi.

Wężel, który nie ma synów, nazywamy liściem lub wężłem zewnętrznym.

Wężel, który nie jest liściem, nazywamy wężłem wewnętrznym.

Własności drzew

Jeśli ostatnią krawędzią drzewa T na ścieżce od korzenia r do wężła x jest (y, x) , to y jest ojcem (rodzicem) wężła x , a x jest synem (dzieckiem) wężła y .

Korzeń drzewa to jedyny wężel, który nie ma ojca.

Dwa wężły mające tego samego ojca nazywamy braćmi.

Wężel, który nie ma synów, nazywamy liściem lub wężłem zewnętrznym.

Wężel, który nie jest liściem, nazywamy wężłem wewnętrznym.

Własności drzew

Stopień wężła x - liczba jego synów.

Głębokość (poziom) wężła x - długość ścieżki od korzenia r do x .

Wysokość wężła x - długość najdłuższej ścieżki od tego wężła w dół do liścia.

Wysokość drzewa T - wysokość korzenia tego drzewa

Własności drzew

Stopień wężła x - liczba jego synów.

Głębokość (poziom) wężła x - długość ścieżki od korzenia r do x .

Wysokość wężła x - długość najdłuższej ścieżki od tego wężła w dół do liścia.

Wysokość drzewa T - wysokość korzenia tego drzewa

Własności drzew

Stopień wężła x - liczba jego synów.

Głębokość (poziom) wężła x - długość ścieżki od korzenia r do x .

Wysokość wężła x - długość najdłuższej ścieżki od tego wężła w dół do liścia.

Wysokość drzewa T - wysokość korzenia tego drzewa

Własności drzew

Stopień wężła x - liczba jego synów.

Głębokość (poziom) wężła x - długość ścieżki od korzenia r do x .

Wysokość wężła x - długość najdłuższej ścieżki od tego wężła w dół do liścia.

Wysokość drzewa T - wysokość korzenia tego drzewa

Własności drzew

Drzewo uporządkowane - drzewo ukorzenione, w którym synowie każdego z węzłów są uporządkowani. Jeśli węzeł ma k synów, to są to kolejno: pierwszy syn, drugi syn, $\dots$, k -ty syn.

Dwa drzewa mogą być różne jako drzewa uporządkowane, ale takie same jako drzewa ukorzenione.

Własności drzew

Drzewo uporządkowane - drzewo ukorzenione, w którym synowie każdego z węzłów są uporządkowani. Jeśli węzeł ma k synów, to są to kolejno: pierwszy syn, drugi syn, $\dots$, k -ty syn. Dwa drzewa mogą być różne jako drzewa uporządkowane, ale takie same jako drzewa ukorzenione.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Drzewo binarne - struktura zdefiniowana na skończonym zbiorze węzłów, która:

- nie zawiera żadnych węzłów, lub
- składa się z trzech rozłącznych zbiorów węzłów:
 - korzenia
 - drzewa binarnego zwanego lewym poddrzewem
 - drzewa binarnego zwanego prawym poddrzewem

Drzewo binarne, które nie zawiera żadnych węzłów, nazywamy drzewem pustym lub drzewem zerowym i oznaczamy stałą *NIL*.

Drzewo binarne

Jeśli lewe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **lewym synem** korzenia drzewa T .

Jeśli prawe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **prawym synem** korzenia drzewa T .

Jeśli lewe lub prawe poddrzewo drzewa binarnego T jest drzewem pustym, to mówimy, że korzeń drzewa T nie ma odpowiedniego lewego lub prawego syna.

Jeśli węzeł drzewa binarnego ma tylko jednego syna, jest istotne, czy jest to syn lewy, czy prawy.

Drzewo binarne

Jeśli lewe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **lewym synem** korzenia drzewa T .

Jeśli prawe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **prawym synem** korzenia drzewa T .

Jeśli lewe lub prawe poddrzewo drzewa binarnego T jest drzewem pustym, to mówimy, że korzeń drzewa T nie ma odpowiedniego lewego lub prawego syna.

Jeśli węzeł drzewa binarnego ma tylko jednego syna, jest istotne, czy jest to syn lewy, czy prawy.

Drzewo binarne

Jeśli lewe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **lewym synem** korzenia drzewa T .

Jeśli prawe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **prawym synem** korzenia drzewa T .

Jeśli lewe lub prawe poddrzewo drzewa binarnego T jest drzewem pustym, to mówimy, że korzeń drzewa T nie ma odpowiedniego lewego lub prawego syna.

Jeśli węzeł drzewa binarnego ma tylko jednego syna, jest istotne, czy jest to syn lewy, czy prawy.

Drzewo binarne

Jeśli lewe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **lewym synem** korzenia drzewa T .

Jeśli prawe pddrzewo drzewa binarnego T jest niepuste, to jego korzeń nazywamy **prawym synem** korzenia drzewa T .

Jeśli lewe lub prawe poddrzewo drzewa binarnego T jest drzewem pustym, to mówimy, że korzeń drzewa T nie ma odpowiedniego lewego lub prawego syna.

Jeśli węzeł drzewa binarnego ma tylko jednego syna, jest istotne, czy jest to syn lewy, czy prawy.

Drzewo binarne

Regularne drzewo binarne - drzewo, w którym każdy węzeł albo jest liściem, albo ma stopień dwa (drzewo binarne, w którym nie ma węzłów posiadających dokładnie jednego syna).

Drzewo pozycyjne - drzewo, w którym synowie tego samego węzła są etykietowani różnymi liczbami całkowitymi dodatnimi. Jeśli nie ma syna etykietowanego liczbą i , mówimy, że brakuje i -tego syna.

Drzewo rzędu k - drzewo pozycyjne, w którym każdy węzeł ma synów etykietowanych wyłącznie liczbami nieprzekraczającymi k .

Drzewo binarne jest drzewem pozycyjnym rzędu $k = 2$.

Drzewo binarne

Regularne drzewo binarne - drzewo, w którym każdy węzeł albo jest liściem, albo ma stopień dwa (drzewo binarne, w którym nie ma węzłów posiadających dokładnie jednego syna).

Drzewo pozycyjne - drzewo, w którym synowie tego samego węzła są etykietowani różnymi liczbami całkowitymi dodatnimi. Jeśli nie ma syna etykietowanego liczbą i , mówimy, że brakuje i -tego syna.

Drzewo rzędu k - drzewo pozycyjne, w którym każdy węzeł ma synów etykietowanych wyłącznie liczbami nieprzekraczającymi k .

Drzewo binarne jest drzewem pozycyjnym rzędu $k = 2$.

Drzewo binarne

Regularne drzewo binarne - drzewo, w którym każdy węzeł albo jest liściem, albo ma stopień dwa (drzewo binarne, w którym nie ma węzłów posiadających dokładnie jednego syna).

Drzewo pozycyjne - drzewo, w którym synowie tego samego węzła są etykietowani różnymi liczbami całkowitymi dodatnimi. Jeśli nie ma syna etykietowanego liczbą i , mówimy, że brakuje i -tego syna.

Drzewo rzędu k - drzewo pozycyjne, w którym każdy węzeł ma synów etykietowanych wyłącznie liczbami nieprzekraczającymi k .

Drzewo binarne jest drzewem pozycyjnym rzędu $k = 2$.

Drzewo binarne

Regularne drzewo binarne - drzewo, w którym każdy węzeł albo jest liściem, albo ma stopień dwa (drzewo binarne, w którym nie ma węzłów posiadających dokładnie jednego syna).

Drzewo pozycyjne - drzewo, w którym synowie tego samego węzła są etykietowani różnymi liczbami całkowitymi dodatnimi. Jeśli nie ma syna etykietowanego liczbą i , mówimy, że brakuje i -tego syna.

Drzewo rzędu k - drzewo pozycyjne, w którym każdy węzeł ma synów etykietowanych wyłącznie liczbami nieprzekraczającymi k .

Drzewo binarne jest drzewem pozycyjnym rzędu $k = 2$.

Drzewo binarne

Pełne drzewo rzędu k - drzewo rzędu k , w którym wszystkie liście mają tę samą głębokość, a wszystkie węzły wewnętrzne mają stopień k , ma następujące własności:

- jeśli wysokość drzew wynosi h , to liczba liści jest równa k^h
- jeśli liczba liści jest równa n , to wysokość drzewa wynosi $\log_k n$
- jeśli wysokość drzewa wynosi h , to liczba węzłów wewnętrznych wynosi $\frac{k^h - 1}{k - 1}$

Wniosek: pełne drzewo binarne o wysokości h ma $2^h - 1$ węzłów wewnętrznych i 2^h liści.

Drzewo binarne

Pełne drzewo rzędu k - drzewo rzędu k , w którym wszystkie liście mają tę samą głębokość, a wszystkie węzły wewnętrzne mają stopień k , ma następujące własności:

- jeśli wysokość drzew wynosi h , to liczba liści jest równa k^h
- jeśli liczba liści jest równa n , to wysokość drzewa wynosi $\log_k n$
- jeśli wysokość drzewa wynosi h , to liczba węzłów wewnętrznych wynosi $\frac{k^h - 1}{k - 1}$

Wniosek: pełne drzewo binarne o wysokości h ma $2^h - 1$ węzłów wewnętrznych i 2^h liści.

Drzewo binarne

Pełne drzewo rzędu k - drzewo rzędu k , w którym wszystkie liście mają tę samą głębokość, a wszystkie węzły wewnętrzne mają stopień k , ma następujące własności:

- jeśli wysokość drzew wynosi h , to liczba liści jest równa k^h
- jeśli liczba liści jest równa n , to wysokość drzewa wynosi $\log_k n$
- jeśli wysokość drzewa wynosi h , to liczba węzłów wewnętrznych wynosi $\frac{k^h - 1}{k - 1}$

Wniosek: pełne drzewo binarne o wysokości h ma $2^h - 1$ węzłów wewnętrznych i 2^h liści.

Drzewo binarne

Pełne drzewo rzędu k - drzewo rzędu k , w którym wszystkie liście mają tę samą głębokość, a wszystkie węzły wewnętrzne mają stopień k , ma następujące własności:

- jeśli wysokość drzew wynosi h , to liczba liści jest równa k^h
- jeśli liczba liści jest równa n , to wysokość drzewa wynosi $\log_k n$
- jeśli wysokość drzewa wynosi h , to liczba węzłów wewnętrznych wynosi $\frac{k^h - 1}{k - 1}$

Wniosek: pełne drzewo binarne o wysokości h ma $2^h - 1$ węzłów wewnętrznych i 2^h liści.

Drzewo binarne

Pełne drzewo rzędu k - drzewo rzędu k , w którym wszystkie liście mają tę samą głębokość, a wszystkie węzły wewnętrzne mają stopień k , ma następujące własności:

- jeśli wysokość drzew wynosi h , to liczba liści jest równa k^h
- jeśli liczba liści jest równa n , to wysokość drzewa wynosi $\log_k n$
- jeśli wysokość drzewa wynosi h , to liczba węzłów wewnętrznych wynosi $\frac{k^h - 1}{k - 1}$

Wniosek: pełne drzewo binarne o wysokości h ma $2^h - 1$ węzłów wewnętrznych i 2^h liści.

Reprezentacja drzewa binarnego

Drzewo binarne reprezentujemy za pomocą struktury wskaźników:

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x)
 - $x.p$ - wskaźnik do ojca węzła x
 - $x.l$ - wskaźnik do lewego syna węzła x
 - $x.r$ - wskaźnik do prawego syna węzła x

Reprezentacja drzewa binarnego

Drzewo binarne reprezentujemy za pomocą struktury wskaźników:

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x)
 - $x.p$ - wskaźnik do ojca węzła x
 - $x.left$ - wskaźnik do lewego syna węzła x
 - $x.right$ - wskaźnik do prawego syna węzła x

Reprezentacja drzewa binarnego

Drzewo binarne reprezentujemy za pomocą struktury wskaźników:

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x)
 - $x.p$ - wskaźnik do ojca węzła x
 - $x.left$ - wskaźnik do lewego syna węzła x
 - $x.right$ - wskaźnik do prawego syna węzła x

Reprezentacja drzewa binarnego

Drzewo binarne reprezentujemy za pomocą struktury wskaźników:

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x)
 - $x.p$ - wskaźnik do ojca węzła x
 - $x.left$ - wskaźnik do lewego syna węzła x
 - $x.right$ - wskaźnik do prawego syna węzła x

Reprezentacja drzewa binarnego

Drzewo binarne reprezentujemy za pomocą struktury wskaźników:

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x)
 - $x.p$ - wskaźnik do ojca węzła x
 - $x.left$ - wskaźnik do lewego syna węzła x
 - $x.right$ - wskaźnik do prawego syna węzła x

Reprezentacja drzewa binarnego

Drzewo binarne reprezentujemy za pomocą struktury wskaźników:

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x)
 - $x.p$ - wskaźnik do ojca węzła x
 - $x.left$ - wskaźnik do lewego syna węzła x
 - $x.right$ - wskaźnik do prawego syna węzła x

Reprezentacja drzewa binarnego

Stałą *NIL* w reprezentacji drzewa binarnego:

- Jeśli $T.root = NIL$, to T jest drzewem pustym
- Jeśli $x.p = NIL$, to x nie ma ojca, czyli jest korzeniem drzewa T
- Jeśli $x.left = NIL$, to x nie ma lewego syna
- Jeśli $x.right = NIL$, to x nie ma prawego syna

Reprezentacja drzewa binarnego

Stałą NIL w reprezentacji drzewa binarnego:

- Jeśli $T.root = NIL$, to T jest drzewem pustym
- Jeśli $x.p = NIL$, to x nie ma ojca, czyli jest korzeniem drzewa T
- Jeśli $x.left = NIL$, to x nie ma lewego syna
- Jeśli $x.right = NIL$, to x nie ma prawego syna

Reprezentacja drzewa binarnego

Stałą NIL w reprezentacji drzewa binarnego:

- Jeśli $T.root = NIL$, to T jest drzewem pustym
- Jeśli $x.p = NIL$, to x nie ma ojca, czyli jest korzeniem drzewa T
- Jeśli $x.left = NIL$, to x nie ma lewego syna
- Jeśli $x.right = NIL$, to x nie ma prawego syna

Reprezentacja drzewa binarnego

Stałą NIL w reprezentacji drzewa binarnego:

- Jeśli $T.root = NIL$, to T jest drzewem pustym
- Jeśli $x.p = NIL$, to x nie ma ojca, czyli jest korzeniem drzewa T
- Jeśli $x.left = NIL$, to x nie ma lewego syna
- Jeśli $x.right = NIL$, to x nie ma prawego syna

Algorytm wypisujący wszystkie węzły

algorytm rekurencyjny wypisujący (w dowolnym porządku) klucze wszystkich węzłów drzewa binarnego T :

```
procedure WYPISZ-KLUCZ-BINARNY( $T$ )
```

```
  BINARY-TREE-WALK( $T$ .root)
```

```
procedure BINARY-TREE-WALK( $x$ )
```

```
  if  $x \neq \text{NIL}$ 
```

```
    write( $x$ .key)
```

```
    BINARY-TREE-WALK( $x$ .left)
```

```
    BINARY-TREE-WALK( $x$ .right)
```

Algorytm wypisujący wszystkie węzły

algorytm rekurencyjny wypisujący (w dowolnym porządku) klucze wszystkich węzłów drzewa binarnego T :

procedure WYPISZ-KLUCZ-BINARNY(T)

 BINARY-TREE-WALK($T.root$)

procedure BINARY-TREE-WALK(x)

if $x \neq \text{NIL}$

 write($x.key$)

 BINARY-TREE-WALK($x.left$)

 BINARY-TREE-WALK($x.right$)

Reprezentacja drzewa uporządkowanego

Drzewo uporządkowane, w którym stopień wężła nie przekracza k , możemy reprezentować na dwa sposoby:

- Zastąpienie pól $x.left$ i $x.right$ polami $x.child_1, x.child_2, \dots, x.child_k$, może prowadzić do marnowania pamięci, jeśli węzły będą miały niepełną liczbę synów.
- Reprezentacja "na lewo syn, na prawo brat" - nie powoduje marnowania pamięci.

Reprezentacja drzewa uporządkowanego

Drzewo uporządkowane, w którym stopień wężła nie przekracza k , możemy reprezentować na dwa sposoby:

- Zastąpienie pól $x.left$ i $x.right$ polami $x.child_1, x.child_2, \dots, x.child_k$, może prowadzić do marnowania pamięci, jeśli węzły będą miały niepełną liczbę synów.
- Reprezentacja "na lewo syn, na prawo brat" - nie powoduje marnowania pamięci.

Reprezentacja drzewa uporządkowanego

Reprezentacja "na lewo syn, na prawo brat":

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x).
 - $x.p$ - wskaźnik do ojca węzła x .
 - $x.l$ - wskaźnik do pierwszego (na lewo) syna węzła x .
 - $x.r$ - wskaźnik do najbliższego brata węzła x (o prawej stronie).

Reprezentacja drzewa uporządkowanego

Reprezentacja "na lewo syn, na prawo brat":

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x).
 - $x.p$ - wskaźnik do ojca węzła x .
 - $x.child$ - wskaźnik do pierwszego (od lewej) syna węzła x .
 - $x.sibling$ - wskaźnik do najbliższego brata węzła x po prawej stronie.

Reprezentacja drzewa uporządkowanego

Reprezentacja "na lewo syn, na prawo brat":

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x).
 - $x.p$ - wskaźnik do ojca węzła x .
 - $x.child$ - wskaźnik do pierwszego (od lewej) syna węzła x .
 - $x.sibling$ - wskaźnik do najbliższego brata węzła x po prawej stronie.

Reprezentacja drzewa uporządkowanego

Reprezentacja "na lewo syn, na prawo brat":

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x).
 - $x.p$ - wskaźnik do ojca węzła x .
 - $x.child$ - wskaźnik do pierwszego (od lewej) syna węzła x .
 - $x.sibling$ - wskaźnik do najbliższego brata węzła x po prawej stronie.

Reprezentacja drzewa uporządkowanego

Reprezentacja "na lewo syn, na prawo brat":

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x).
 - $x.p$ - wskaźnik do ojca węzła x .
 - $x.child$ - wskaźnik do pierwszego (od lewej) syna węzła x .
 - $x.sibling$ - wskaźnik do najbliższego brata węzła x po prawej stronie.

Reprezentacja drzewa uporządkowanego

Reprezentacja "na lewo syn, na prawo brat":

- Drzewo T posiada atrybut $T.root$ wskazuje na korzeń tego drzewa.
- Każdy węzeł x drzewa T jest rekordem posiadającym następujące pola:
 - $x.key$ - klucz (wartość przechowywana przez węzeł x).
 - $x.p$ - wskaźnik do ojca węzła x .
 - $x.child$ - wskaźnik do pierwszego (od lewej) syna węzła x .
 - $x.sibling$ - wskaźnik do najbliższego brata węzła x po prawej stronie.

Algorytm wypisujący wszystkie węzły

algorytm rekurencyjny wypisujący (w dowolnym porządku) klucze wszystkich węzłów drzewa binarnego T :

```
procedure WYPISZ-KLUCZ-LR( $T$ )
```

```
  BINARY-TREE-WALK( $T$ .root)
```

```
procedure LR-TREE-WALK( $x$ )
```

```
  if  $x \neq \text{NIL}$ 
```

```
    write( $x$ .key)
```

```
    BINARY-TREE-WALK( $x$ .child)
```

```
    BINARY-TREE-WALK( $x$ .sibling)
```

Algorytm wypisujący wszystkie węzły

algorytm rekurencyjny wypisujący (w dowolnym porządku) klucze wszystkich węzłów drzewa binarnego T :

procedure WYPISZ-KLUCZ-LR(T)

 BINARY-TREE-WALK(T .root)

procedure LR-TREE-WALK(x)

if $x \neq \text{NIL}$

 write(x .key)

 BINARY-TREE-WALK(x .child)

 BINARY-TREE-WALK(x .sibling)

Drzewa decyzyjne

Drzewo decyzyjne - graficzna metoda reprezentacji i wspomagania procesu decyzyjnego.

Wewnętrzny węzeł drzewa decyzyjnego reprezentuje sprawdzanie pewnego warunku. Na podstawie otrzymanego wyniku wybierana jest jedna krawędź prowadząca do synów tego węzła.

Liść drzewa decyzyjnego przedstawia podjęcie określonej decyzji.

Proces decyzyjny polega na przejściu drzewa od korzenia do jednego z liści.

Drzewa decyzyjne

Drzewo decyzyjne - graficzna metoda reprezentacji i wspomagania procesu decyzyjnego.

Wewnętrzny węzeł drzewa decyzyjnego reprezentuje sprawdzanie pewnego warunku. Na podstawie otrzymanego wyniku wybierana jest jedna krawędź prowadząca do synów tego węzła.

Liść drzewa decyzyjnego przedstawia podjęcie określonej decyzji. Proces decyzyjny polega na przejściu drzewa od korzenia do jednego z liści.

Drzewa decyzyjne

Drzewo decyzyjne - graficzna metoda reprezentacji i wspomagania procesu decyzyjnego.

Wewnętrzny węzeł drzewa decyzyjnego reprezentuje sprawdzanie pewnego warunku. Na podstawie otrzymanego wyniku wybierana jest jedna krawędź prowadząca do synów tego węzła.

Liść drzewa decyzyjnego przedstawia podjęcie określonej decyzji.

Proces decyzyjny polega na przejściu drzewa od korzenia do jednego z liści.

Drzewa decyzyjne

Drzewo decyzyjne - graficzna metoda reprezentacji i wspomagania procesu decyzyjnego.

Wewnętrzny węzeł drzewa decyzyjnego reprezentuje sprawdzanie pewnego warunku. Na podstawie otrzymanego wyniku wybierana jest jedna krawędź prowadząca do synów tego węzła.

Liść drzewa decyzyjnego przedstawia podjęcie określonej decyzji.

Proces decyzyjny polega na przejściu drzewa od korzenia do jednego z liści.

Drzewa decyzyjne -zastosowanie

Przykładowe zastosowanie drzew decyzyjnych:

- diagnostyka medyczna
- przyznawanie kredytu
- diagnozowanie sprzętu komputerowego

Wzrostowy algorytm

Wzrostowe algorytmy (wzrostowe drzewo decyzyjne)

przykład(ów)

Drzewa decyzyjne -zastosowanie

Przykładowe zastosowanie drzew decyzyjnych:

- diagnostyka medyczna
- przyznawanie kredytu
- diagnozowanie sprzętu komputerowego
- schematy negocjacji
- uczenie maszynowe (wyodrębnianie wiedzy z zestawu przykładów)

Drzewa decyzyjne -zastosowanie

Przykładowe zastosowanie drzew decyzyjnych:

- diagnostyka medyczna
- przyznawanie kredytu
- diagnozowanie sprzętu komputerowego
- schematy negocjacji
- uczenie maszynowe (wyodrębnianie wiedzy z zestawu przykładów)

Drzewa decyzyjne -zastosowanie

Przykładowe zastosowanie drzew decyzyjnych:

- diagnostyka medyczna
- przyznawanie kredytu
- diagnozowanie sprzętu komputerowego
- schematy negocjacji
- uczenie maszynowe (wyodrębnianie wiedzy z zestawu przykładów)

Drzewa decyzyjne -zastosowanie

Przykładowe zastosowanie drzew decyzyjnych:

- diagnostyka medyczna
- przyznawanie kredytu
- diagnozowanie sprzętu komputerowego
- schematy negocjacji
- uczenie maszynowe (wyodrębnianie wiedzy z zestawu przykładów)

Drzewa decyzyjne -zastosowanie

Przykładowe zastosowanie drzew decyzyjnych:

- diagnostyka medyczna
- przyznawanie kredytu
- diagnozowanie sprzętu komputerowego
- schematy negocjacji
- uczenie maszynowe (wyodrębnianie wiedzy z zestawu przykładów)