Universitatea Politehnica din București
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare

ARBORI PARȚIALI

Definiții

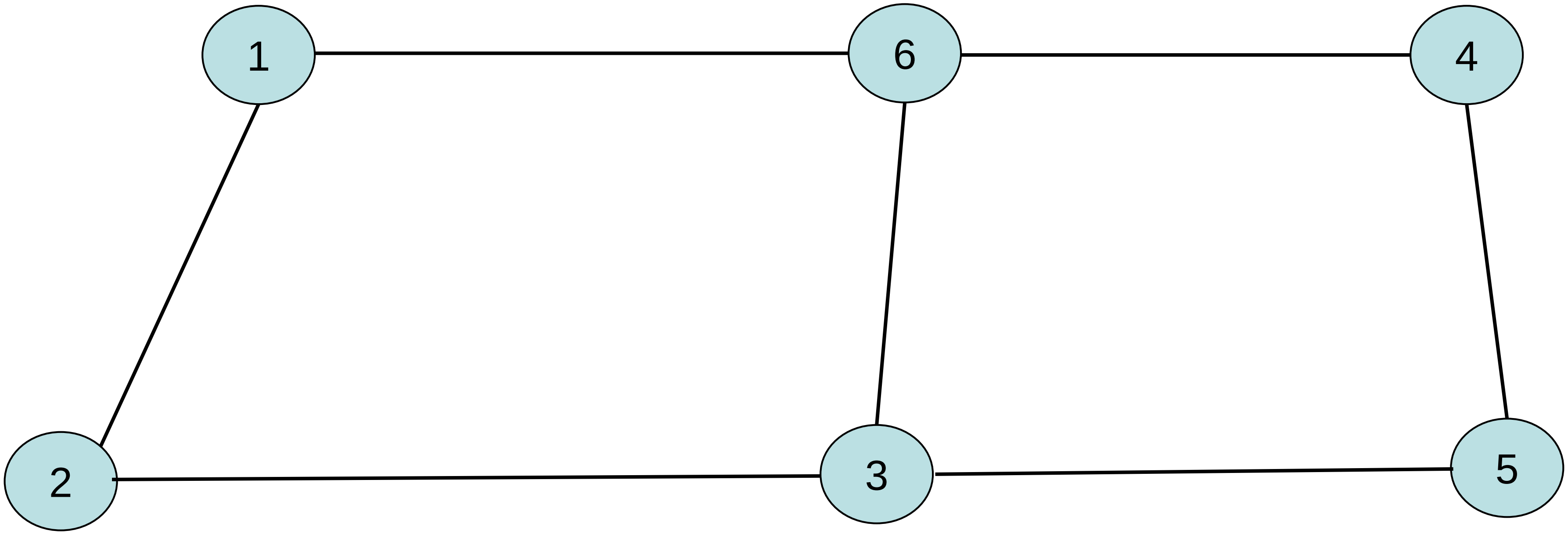
- Un graf neorientat conex și aciclic se numește **arbore**
- Un graf neorientat aciclic și neconex se numește **pădure**

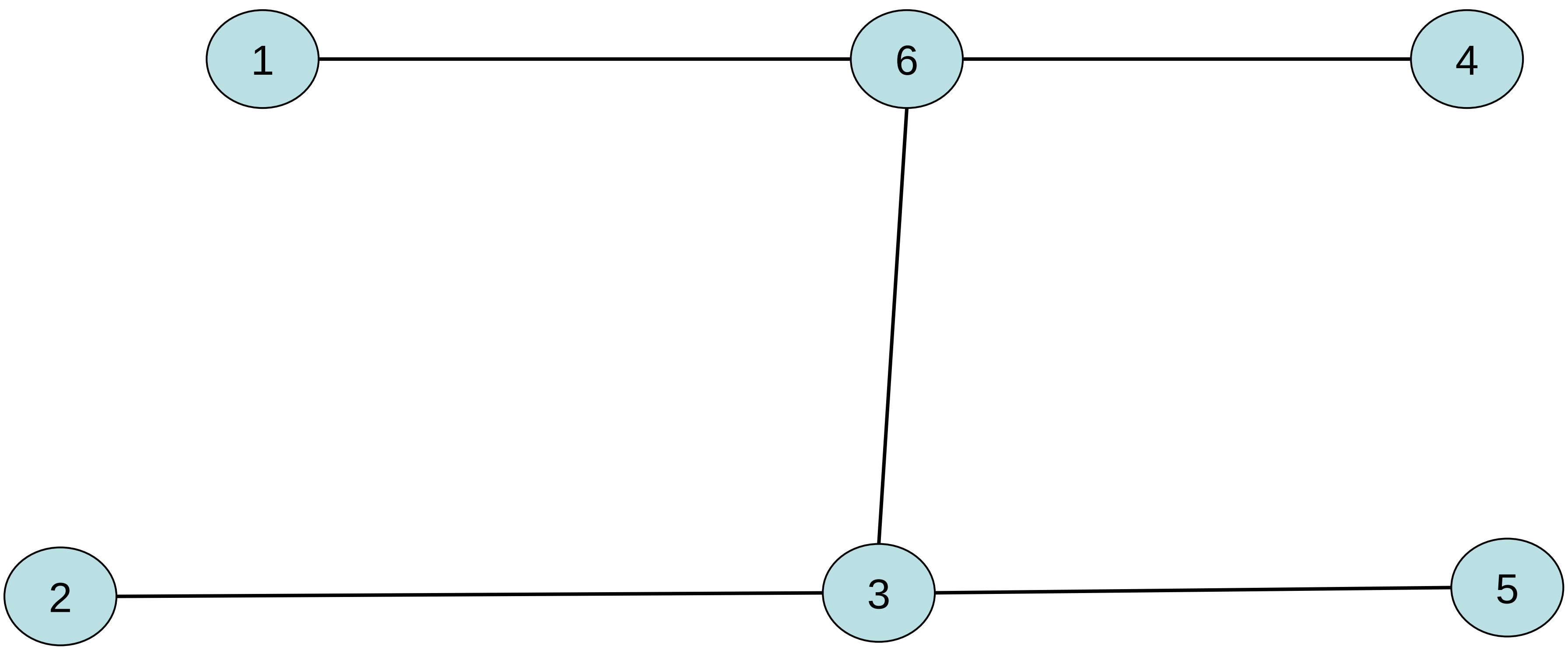
Observație

- Fiecare componentă conexă a unui graf aciclic neconex (pădure) este un arbore

Arbore parțial

- Un graf parțial conex și aciclic al unui graf neorientat se numește **arbore parțial**


1
6
4
2
3
5


1
6
4
2
3
5

Teorema 1

- Condiția necesară și suficientă ca un graf să conțină cel puțin un **arbore parțial** este ca graful să fie **conex**

Demonstrație - Necesitatea

- Presupunem că graful admite un arbore parțial
- Arborele parțial este conex și adăugând muchiile care sunt în graf, dar nu sunt în arborele parțial, el rămâne conex
- Deci graful este conex

Demonstrație – Suficiența

- Presupunem că graful este conex
- Dacă graful este conex minimal, el este arborele parțial căutat
- Altfel, există o muchie `(x,y)`, astfel încât graful parțial G_1 obținut prin eliminarea muchiei `(x,y)` este conex
- Dacă G_1 este conex minimal, arborele parțial căutat este G_1, altfel continuăm procedeul de eliminare a muchiilor, până când obținem un graf conex minimal, care va fi arborele parțial căutat

Teorema 2

- Fie `G` un graf conex cu `n` vârfuri și `m` muchii
- Numărul de muchii ce trebuie eliminate pentru a obține un arbore parțial este `m-n+1` (acesta se numește **numărul ciclomatic** al grafului)

Demonstrație

- Presupunem că prin eliminarea unui număr oarecare de muchii din G am obținut un graf G' fără cicluri (o pădure)
- Fiecare dintre componentele conexe ale lui G' este un arbore
- Să notăm cu p numărul componentelor conexe, cu n_i numărul de vârfuri din componenta conexă i, unde $i \in \{1, 2, \ldots, p\}$ și cu m_i numărul de muchii din componenta conexă i, $i \in \{1, 2, \ldots, p\}$

Demonstrație

- Evident că $m_i = n_i - 1$, $\forall i \in \{1, 2, \dots, p\}$
- Numărul de muchii din G' este:
- $(n_1-1)+(n_2-1)+...+(n_p-1)=n-p$
- Deci au fost eliminate $m-n+p$ muchii
- Când G' este arbore, deci conex ($p=1$), numărul muchiilor eliminate este $m-n+1$

Arbori parțiali obținuți prin parcurgerea grafurilor

- Prin parcurgerea unui graf neorientat conex, fiecare vârf din graf va fi vizitat o singură dată
- Exceptând vârful de start, pentru vizitarea fiecărui vârf se utilizează o singură muchie a grafului
- În total pentru parcurgere au fost utilizate `n-1` muchii – denumite **muchiile arborelui** (*tree edges*) – care nu formează cicluri, deci care constituie un arbore parțial al grafului dat

Observații

- În cazul în care graful parcurs nu este conex, parcurgerea se repetă pentru fiecare componentă conexă, obținându-se o **pădure**, formată din arborii parțiali corespunzători fiecărei componente conexe
- Prin parcurgerea unui graf orientat se obține o **arborescență** (graf orientat fără circuite, în care fiecare vârf este accesibil din vârful de start)

Clasificarea muchiilor

- Prin parcurgere, muchiile grafului pot fi clasificate în 4 categorii:
- **1**. **Muchii ale arborelui** (*tree edges*) – muchiile arborelui parțial; `(x,y)` este muchie a arborelui dacă și numai dacă vârful `y` a fost vizitat explorând muchia `(x,y)`

Clasificarea muchiilor

- **2**. **Muchii de revenire** (*back edges*) – muchii care nu aparțin arborelui parțial și care conectează un vârf de un strămoș al său în arborele parțial, adică muchii de forma `(x,y)` care conectează pe `x` de un vârf `y` aflat în lanțul de la vârful de start la `x`, mai aproape de vârful de start decât `x`

Clasificarea muchiilor

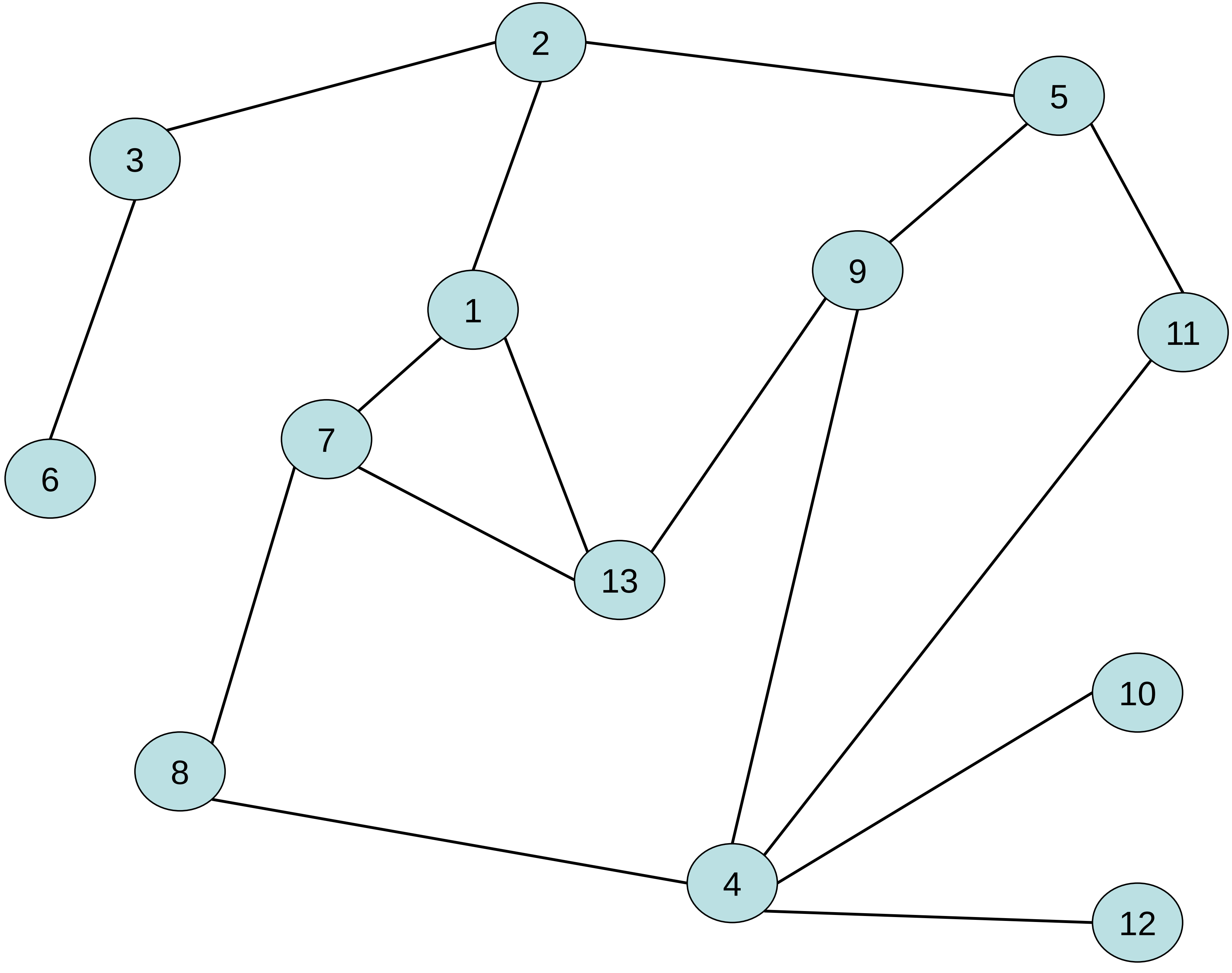
- **3**. **Muchii de înaintare** (*forward edges*) – muchii care nu aparțin arborelui parțial și care conectează un vârf de un descendent al său în arborele parțial, adică o muchie de forma `(x,y)` care conectează pe `x` de un vârf `y`, astfel încât `x` se află în lanțul de la vârful de start la `y` mai aproape de vârful de start decât `y`

Clasificarea muchiilor

- **4. Muchii de traversare** (*cross edges*) – toate celelalte muchii

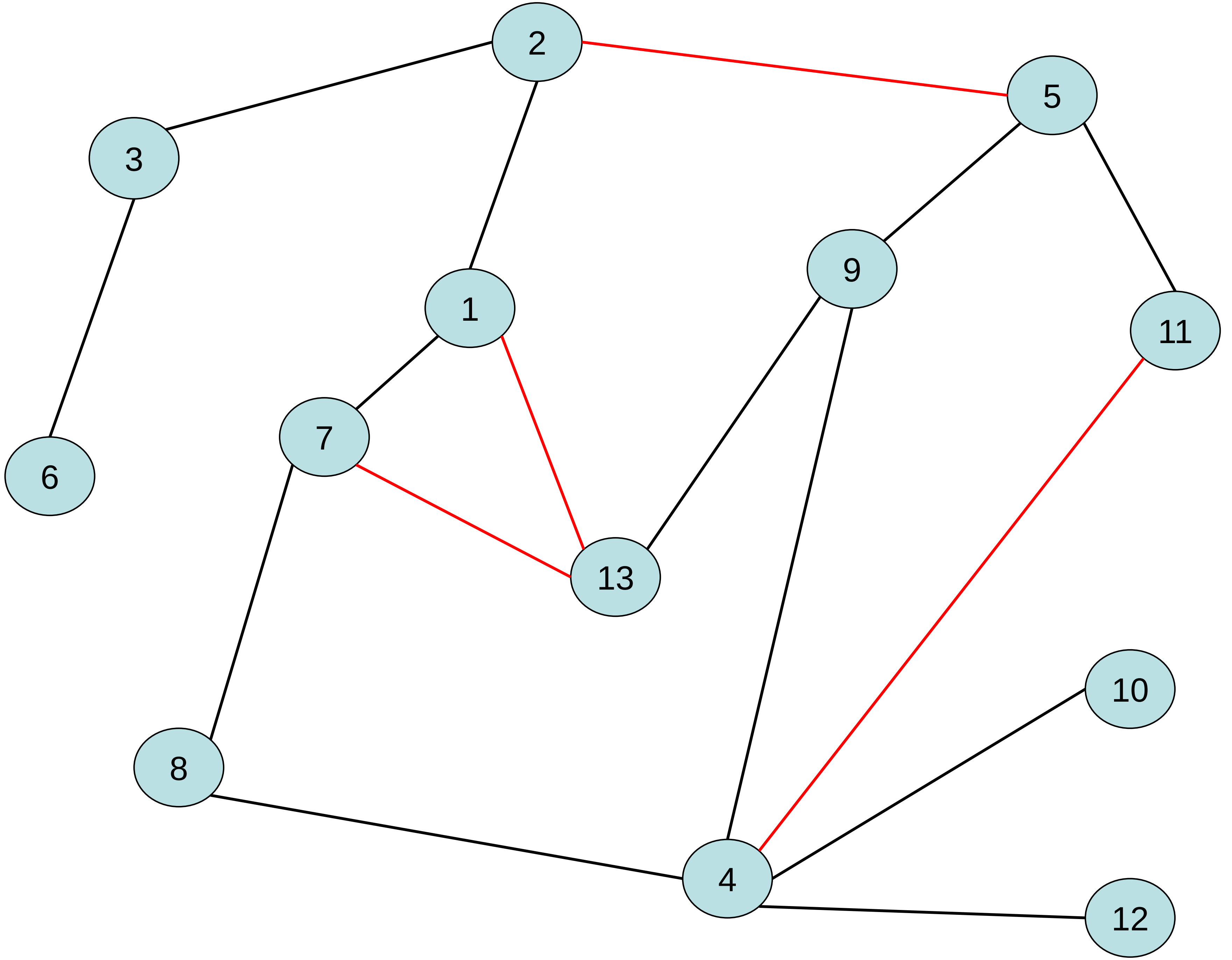
Exemplu

- Se consideră graful neorientat conex


2
5
3
9
1
11
7
6
13
10
8
4
12

Exemplu

- Se parcurge în adâncime graful, începând din vârful 2
- Muchiile negre reprezintă muchiile din arborele parțial obținut prin parcurgerea în adâncime (*tree edges*)
- Muchiile roșii reprezintă muchiile de revenire (*back edges*)

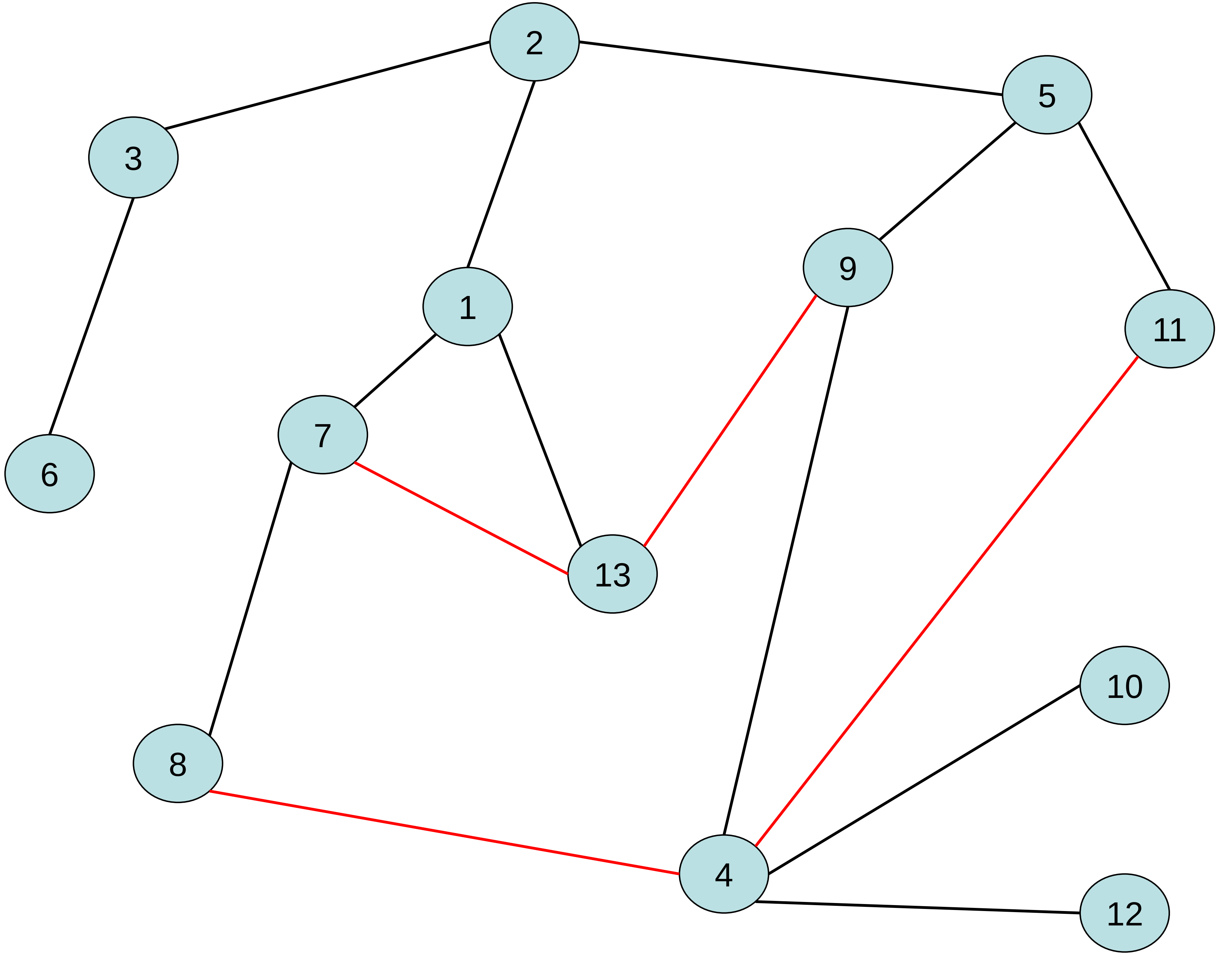

2
5
3
9
1
11
7
6
13
10
8
4
12

Observații

- Un graf (orientat sau neorientat) este aciclic dacă nu conține muchii de revenire (*back edges*)
- Prin parcurgerea în adâncime a unui graf neorientat, muchiile pot fi clasificate doar în muchii ale arborelui parțial DFS (*tree edges*) sau muchii de revenire (*back edges*)

Exemplu

- Se parcurge în lățime graful, începând din vârful 2
- Muchiile negre reprezintă muchiile din arborele parțial obținut prin parcurgerea în lățime (*tree edges*)
- Muchiile roșii reprezintă muchiile de traversare (*cross edges*)


2
5
3
9
1
11
7
6
13
10
8
4
12

Observații

- Pentru orice vârf `x` din graf, lanțul unic care unește vârful de start de `x` în arborele parțial BFS reprezintă un lanț cu număr minim de muchii de la vârful de start la `x` în graf
- Lungimea acestui lanț se numește distanța de la vârful de start la `x` și se notează `d(x)`

Observații

- Prin parcurgerea BFS a unui graf neorientat, muchiile pot fi clasificate doar în două categorii: muchii ale arborelui parțial BFS (*tree edges*) și muchii de traversare (*cross edges*)
- Dacă `(x,y)` este o muchie a arborelui BFS, atunci `d(y)=d(x)+1`
- Dacă `(x,y)` este o muchie de traversare, atunci `d(x)=d(y)` sau `d(y)=d(x)+1`