

Proiectarea și Analiza Algoritmilor

S II - 2025

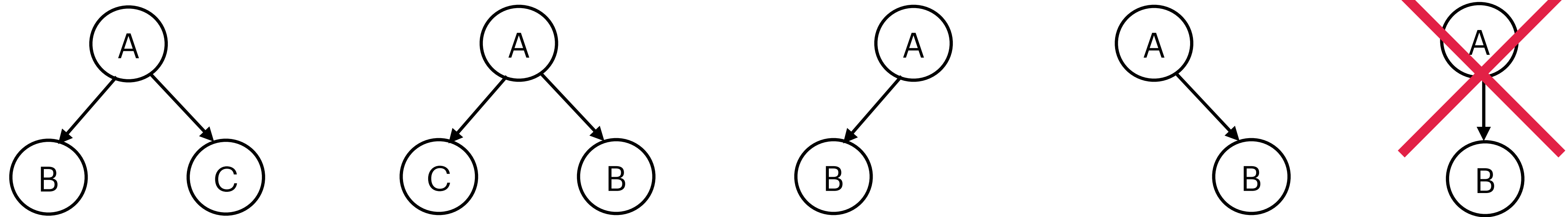
Curs 3

Arbori Binari. Arbori Binari Ordonati

Definiții

Prin **arbore binar** se înțelege o mulțime de $n \geq 0$ noduri care dacă nu este vidă, conține un anumit nod numit **rădăcină**, iar restul nodurilor formează doi arbori binari disjuncți numiți **subarbore stâng** respectiv **subarbore drept**.

Exemple:



Cei 4 arbori binari prezentați sunt considerați arbori diferiți

Se observă că arborii binari sunt un subset al arborilor generalizați. Tot ce se aplică pentru arbori generalizați (teorie, definiții, implementare, etc..) rămâne valabil la arborii binari. Orice arbore binar poate fi tratat ca un arbore generalizat. Restricțiile specifice arborilor binari ne oferă posibilitatea pentru noi tehnici de implementare care sunt mai eficiente decât tehnicile de implementare ale arborilor generalizați.

Structura **arbore binar** este importantă deoarece:

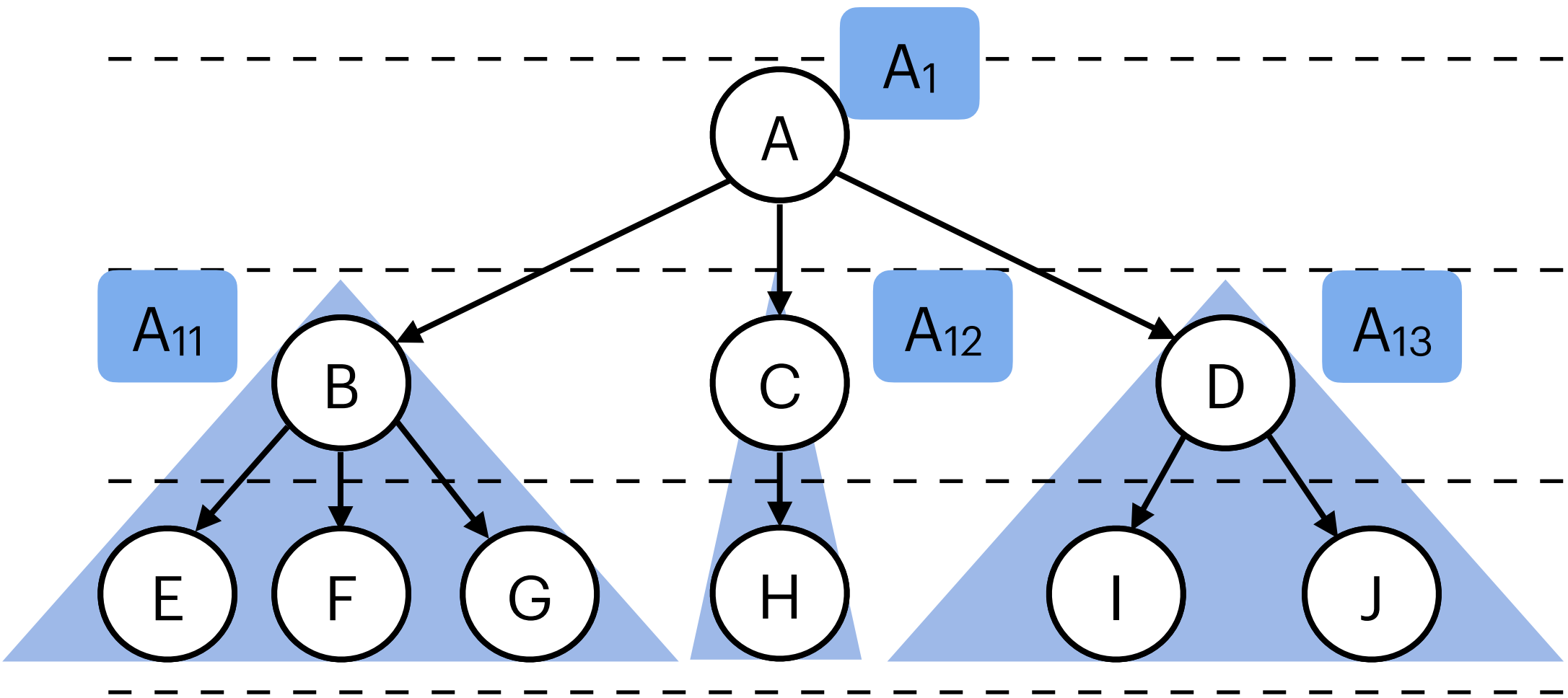
- Este simplu de **reprezentat**

- Este ușor de **prelucrat**, bucurându-se de o serie de proprietăți specifice

- Orice** structură de **arbore** poate fi transformată într-o structură **arbore binar**

Transformarea AGen în ABin

Fie un arbore generalizat oarecare A, care are rădăcina A_1 și subarborii $A_{11}, A_{12}, \dots, A_{1k}$



	0	1	2	3	4	5	6	7	8	9	10
Val	–	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Reprezentarea prin tablouri, prim fiu - frate dreapta

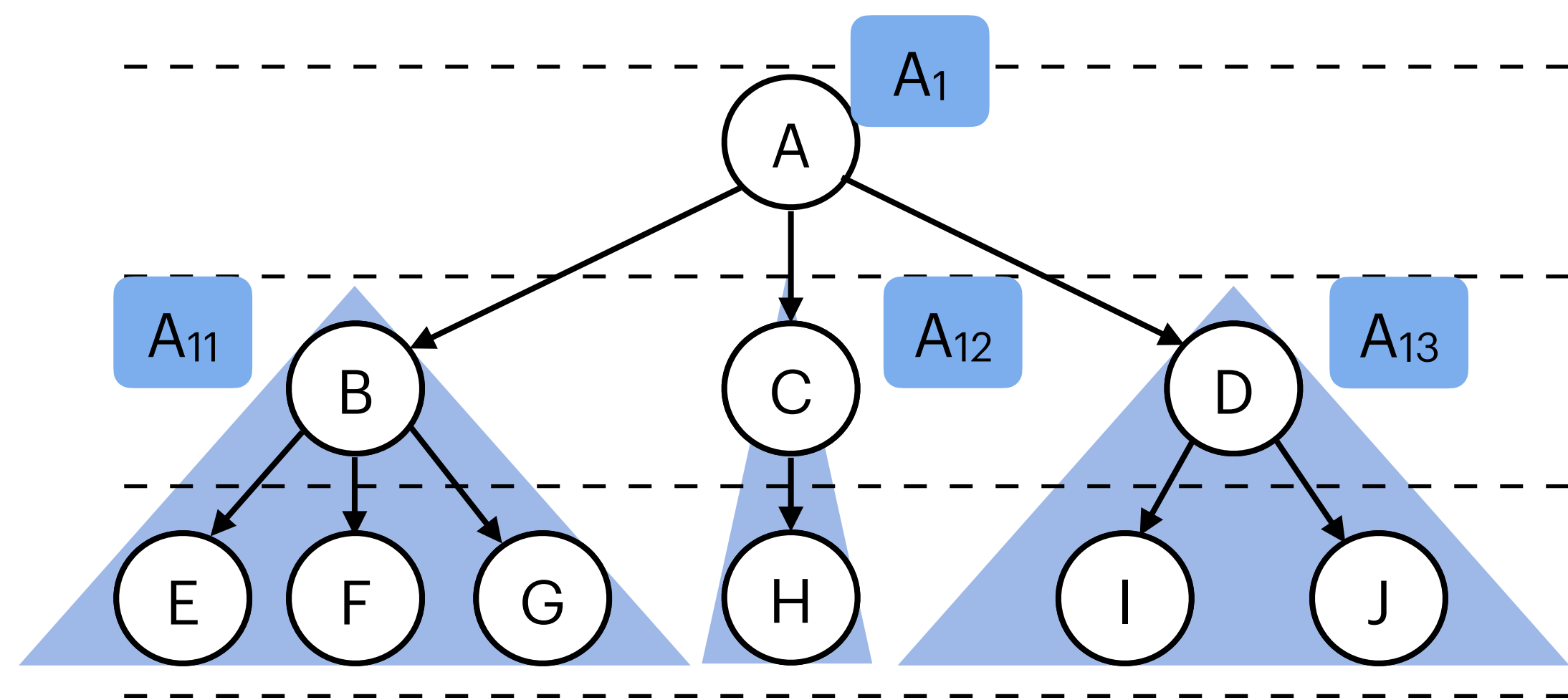
Se ia A_1 drept rădăcină a arborelui binar

Se face subarborele A_{11} fiul său stâng

Fiecare subarbore A_{1i} se face fiul drept a lui $A_{1, i-1}$ pentru $2 \leq i \leq k$

Se continuă în aceeași manieră transformând după același algoritm fiecare din subarborii rezultați, până la parcurgerea integrală a arborelui inițial

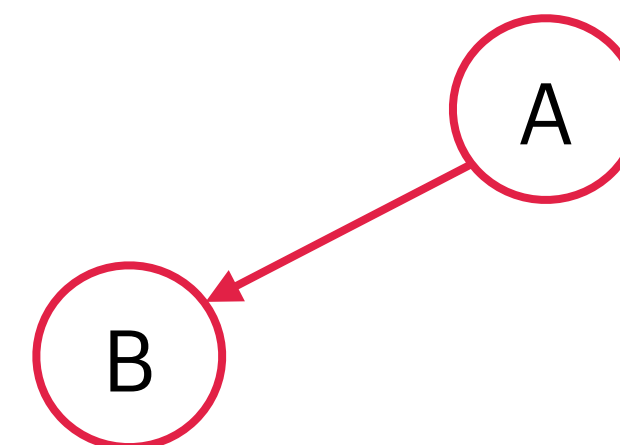
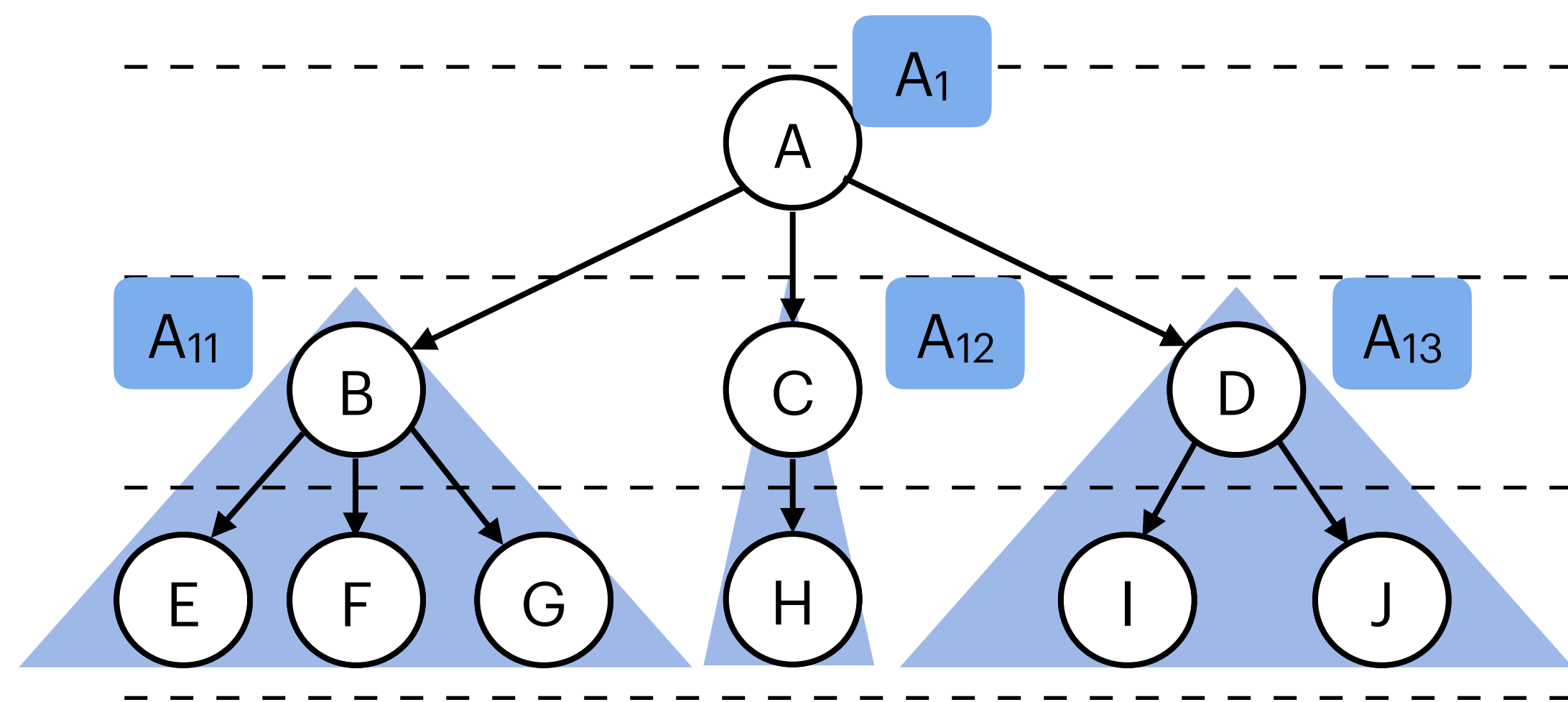
Transformarea AGen în ABin



A

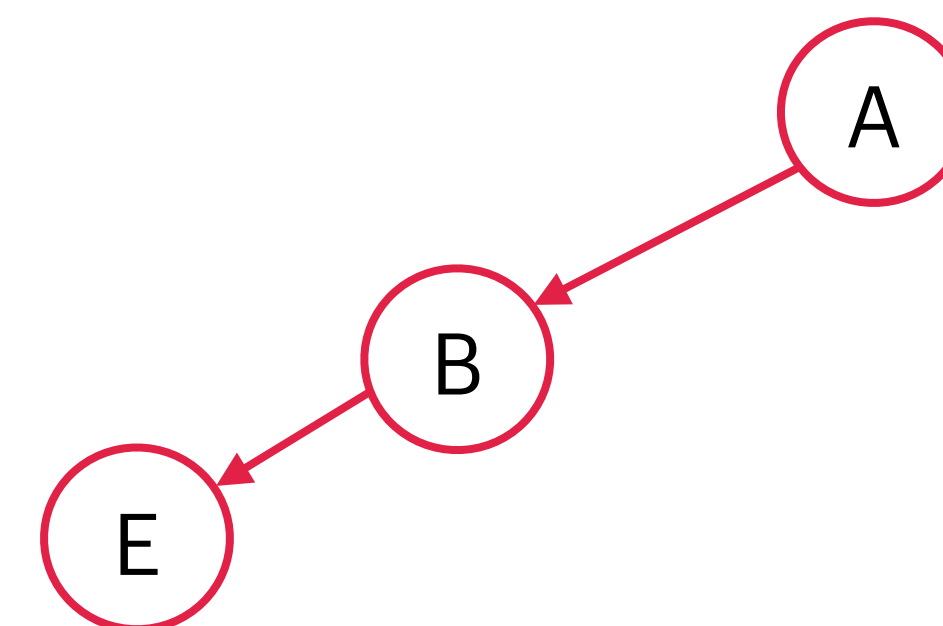
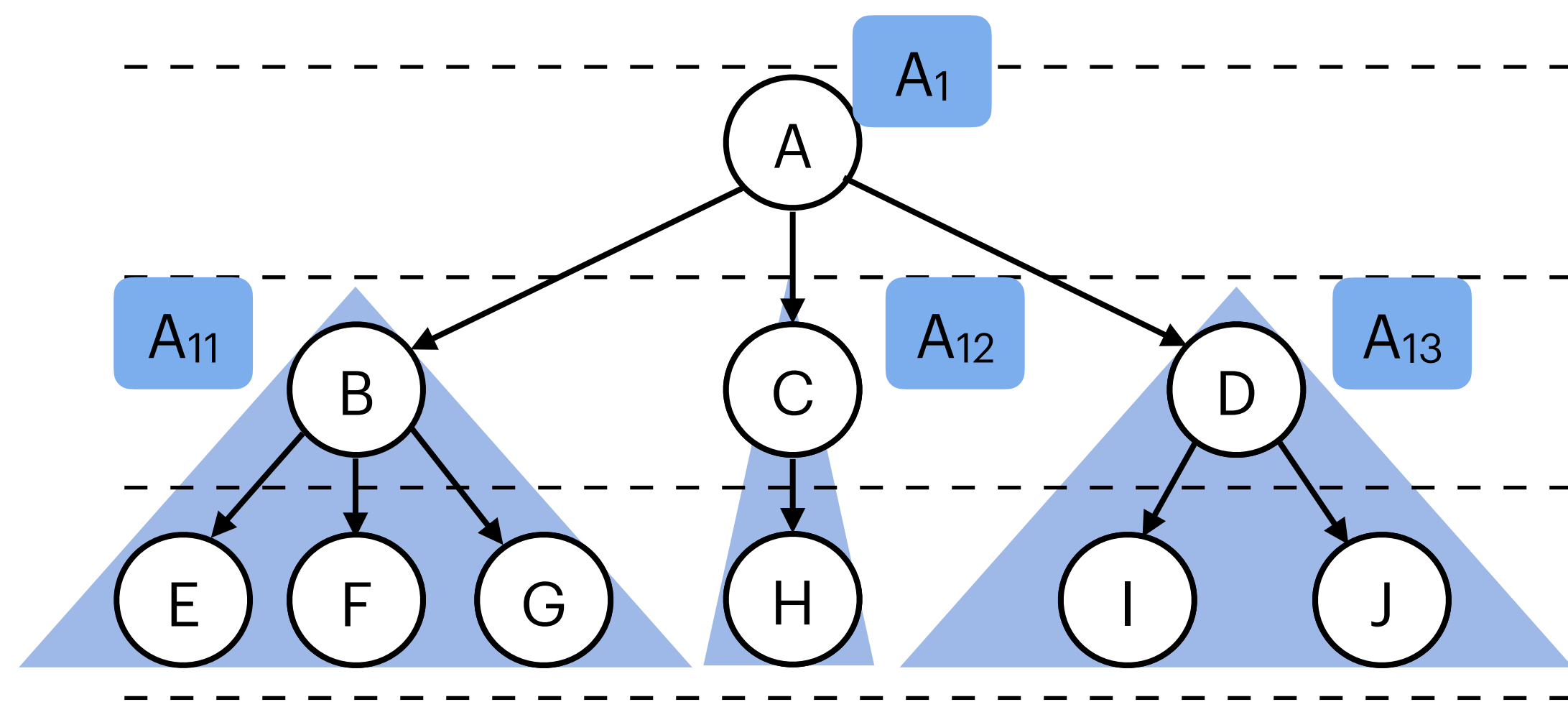
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



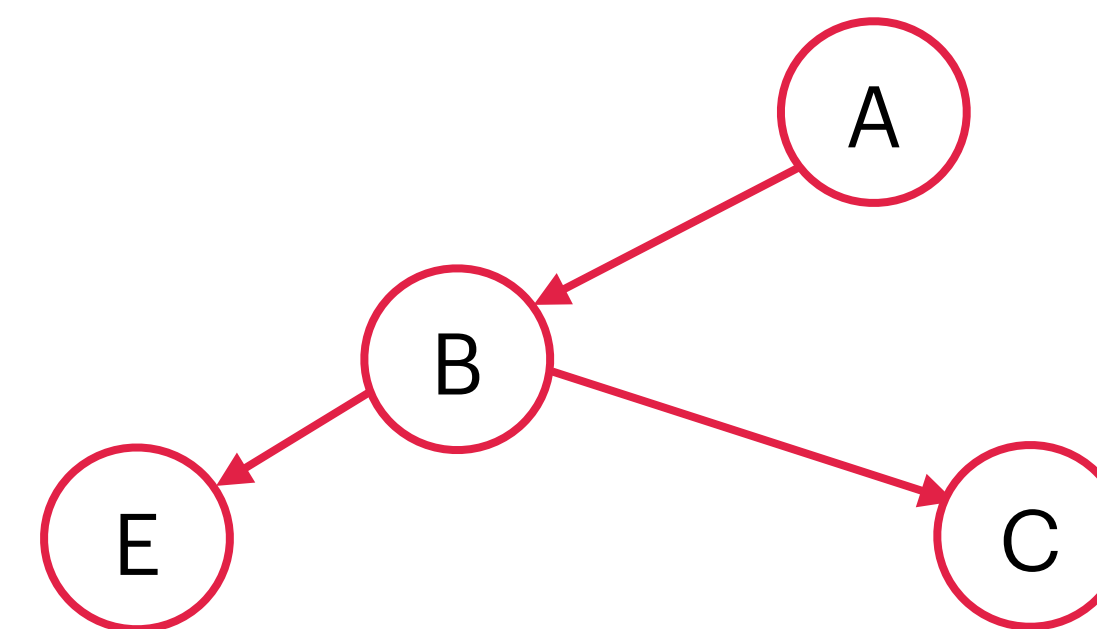
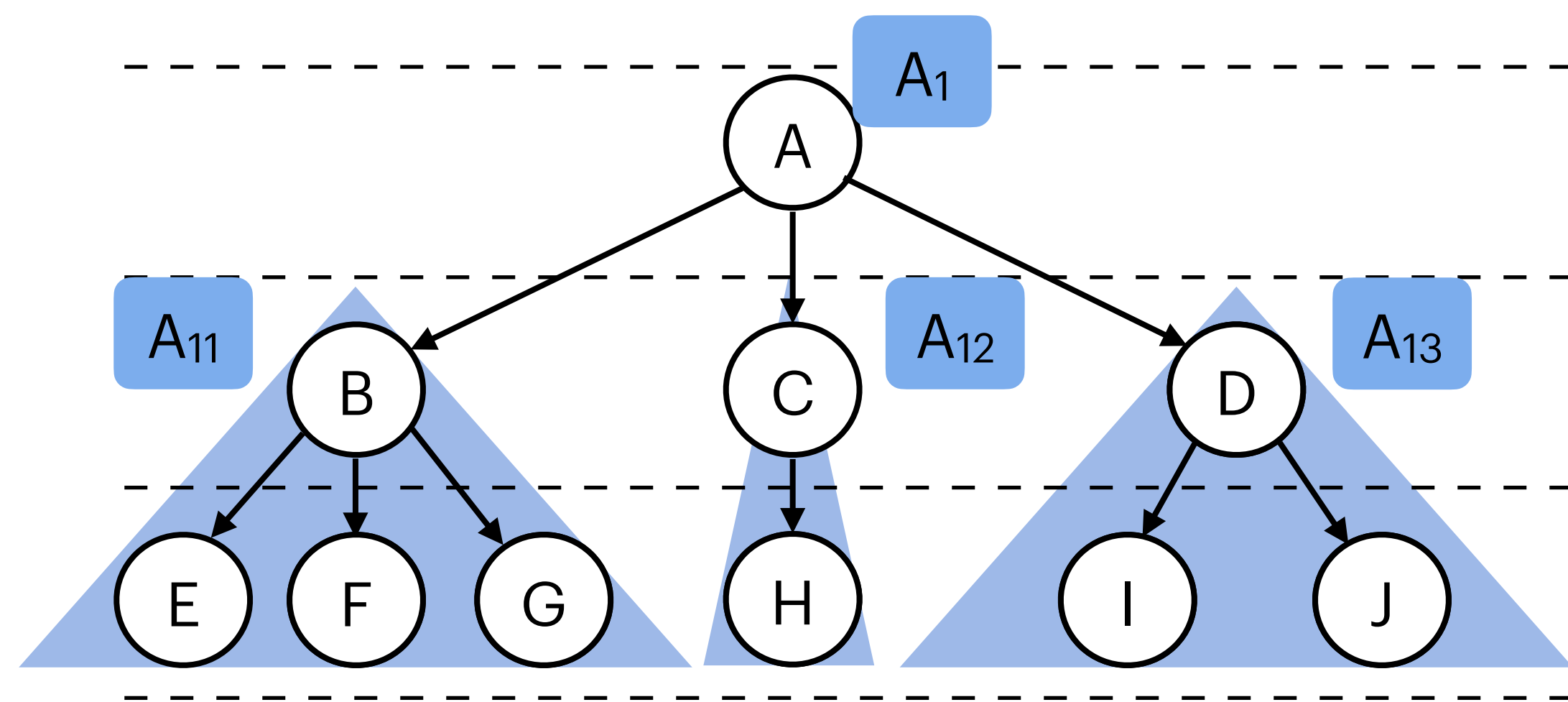
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



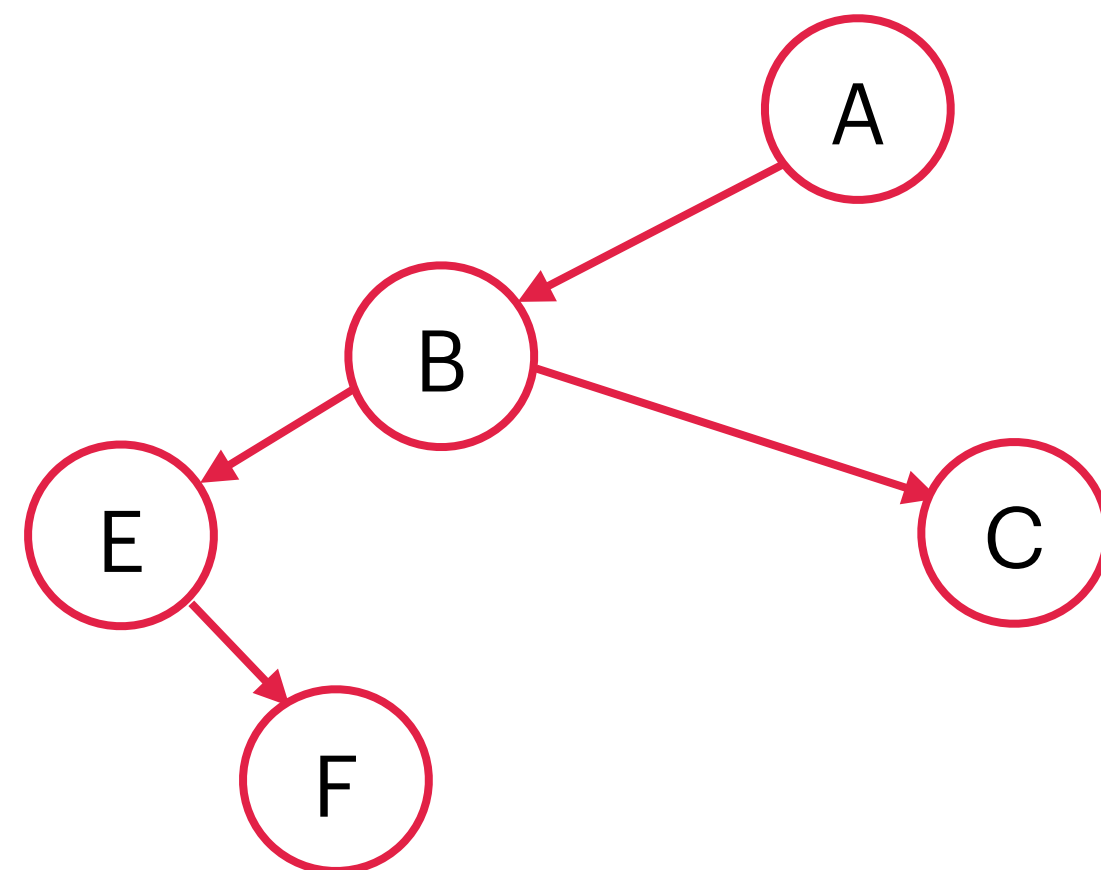
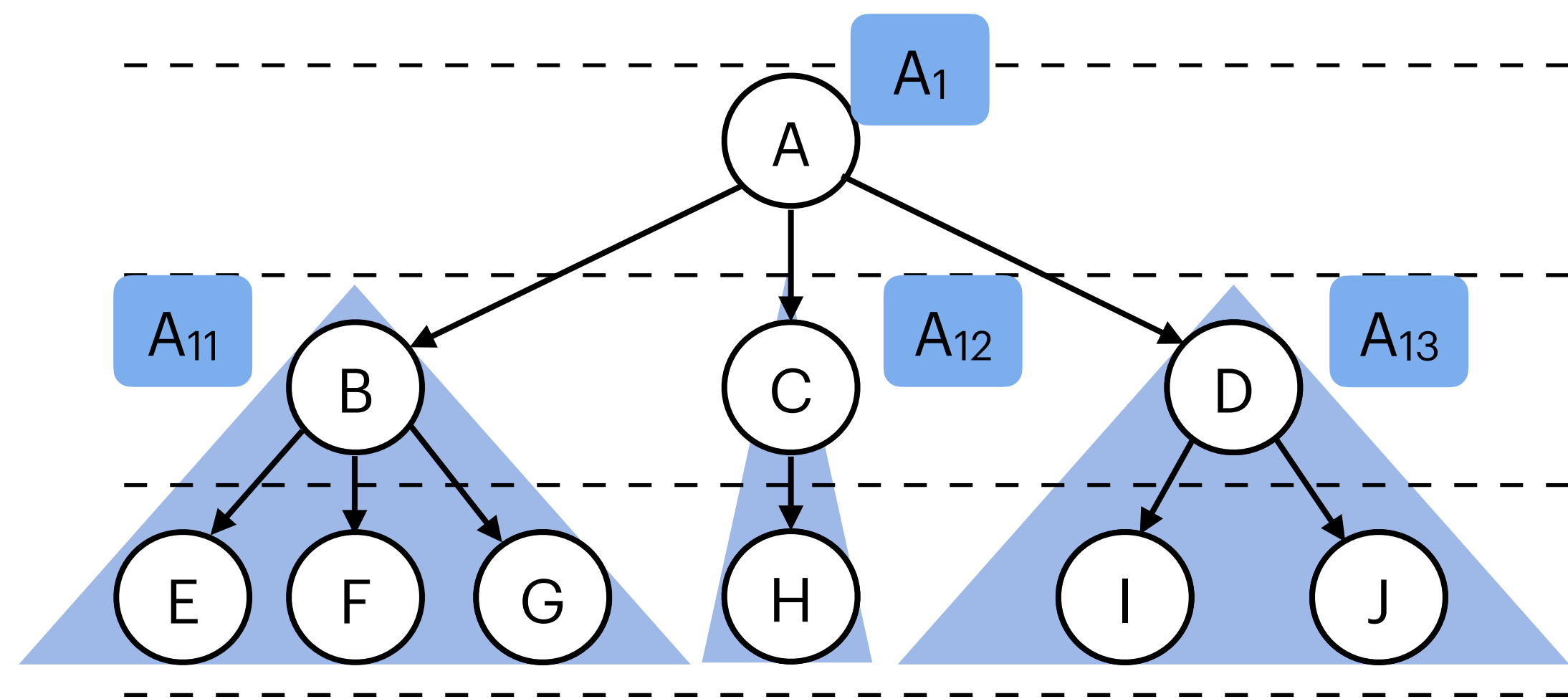
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



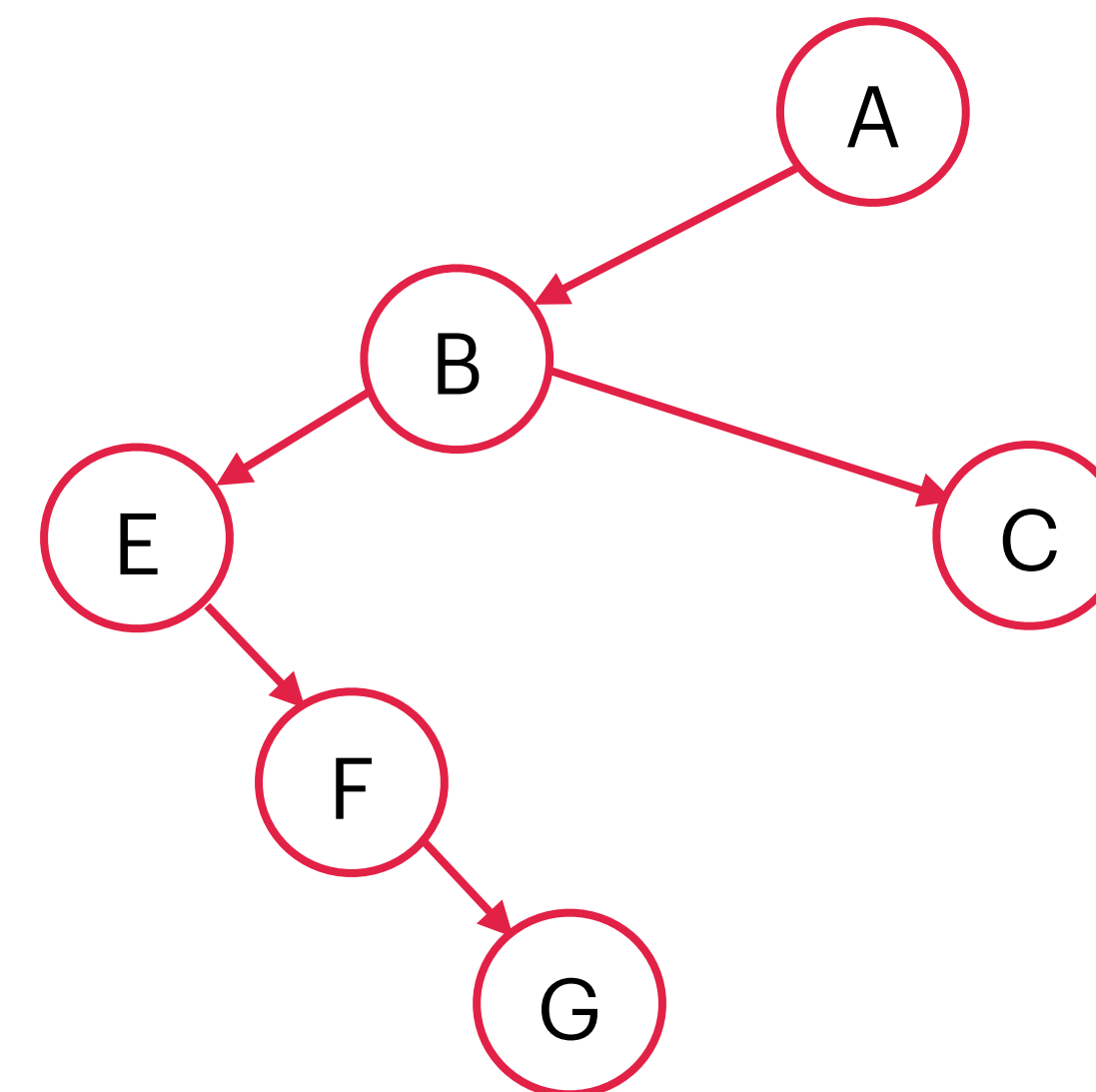
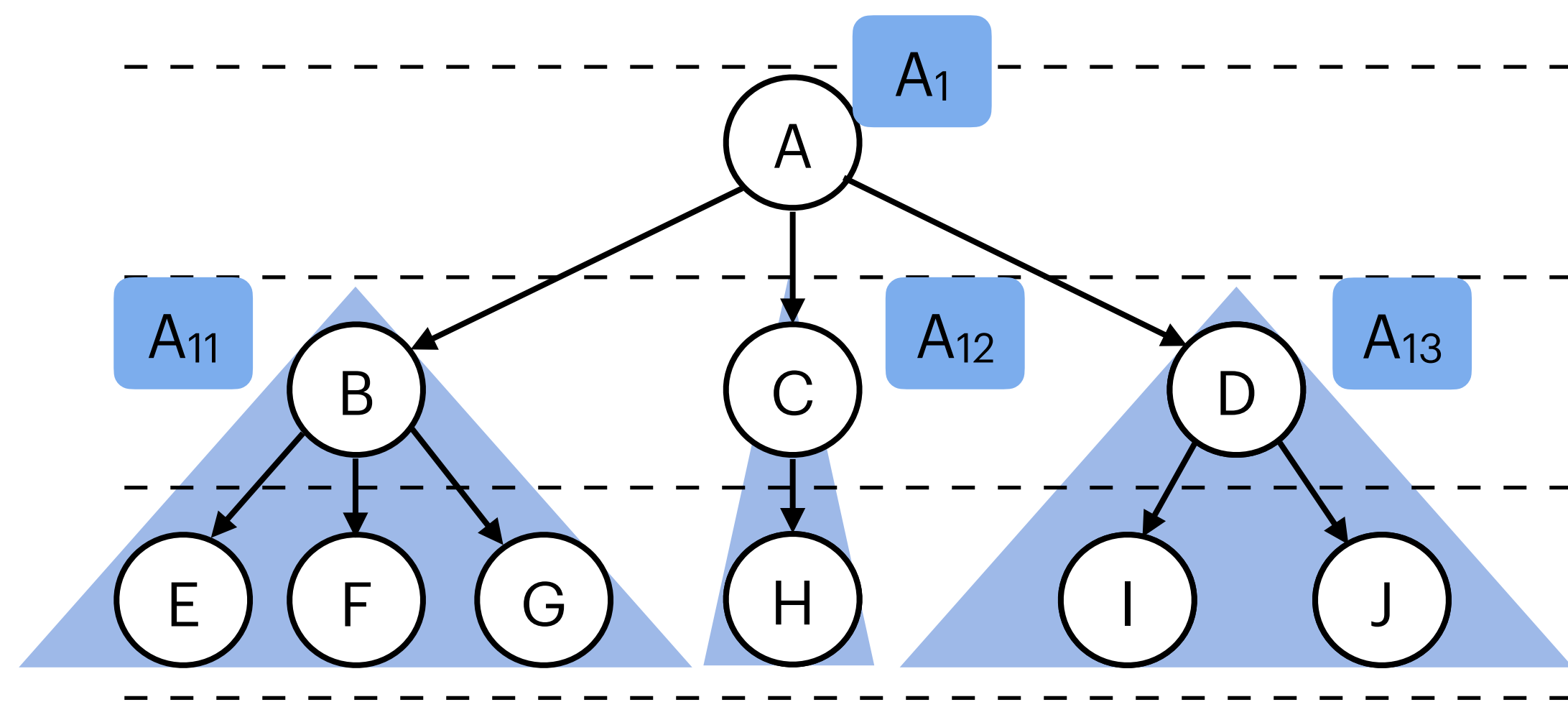
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



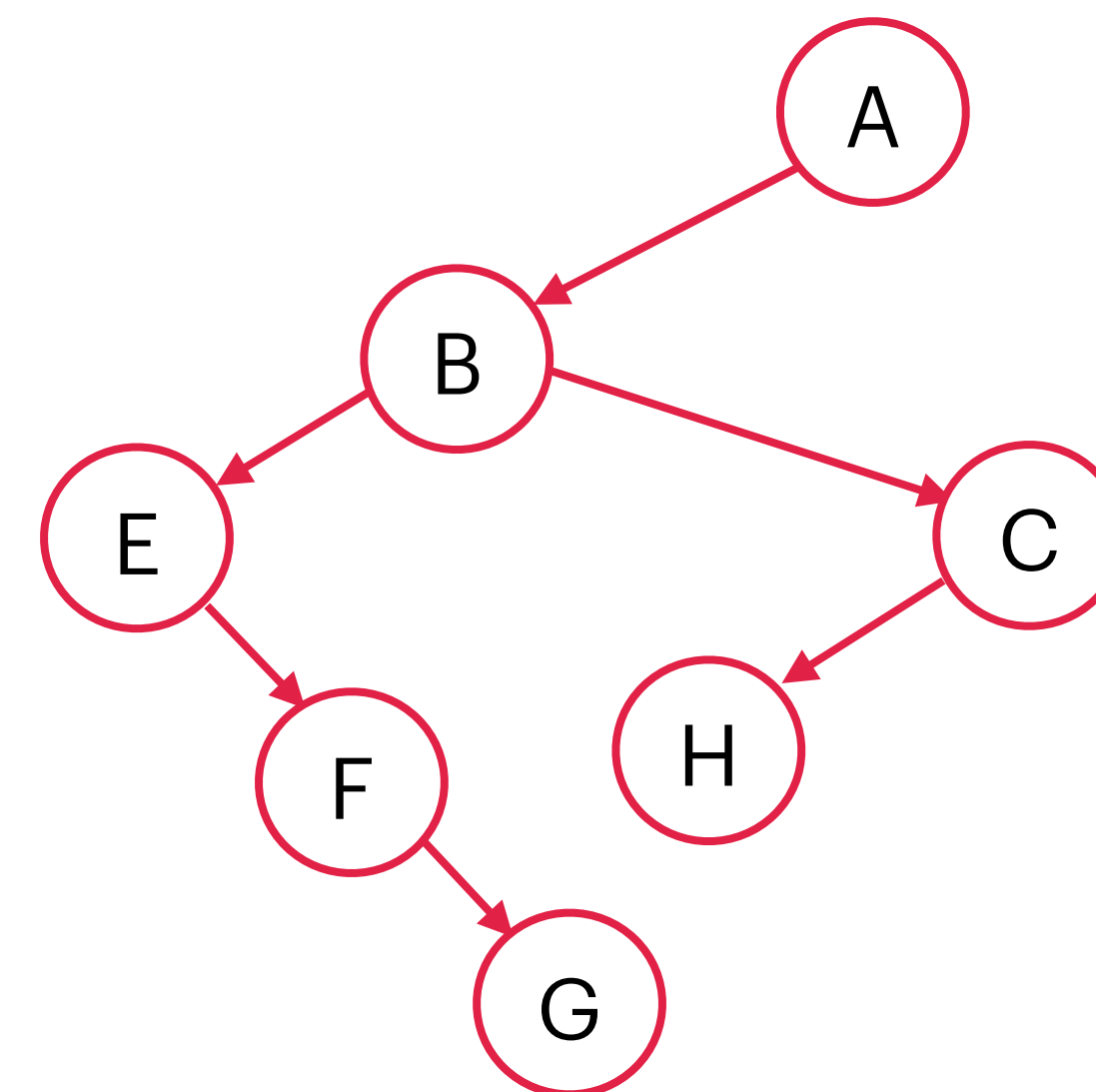
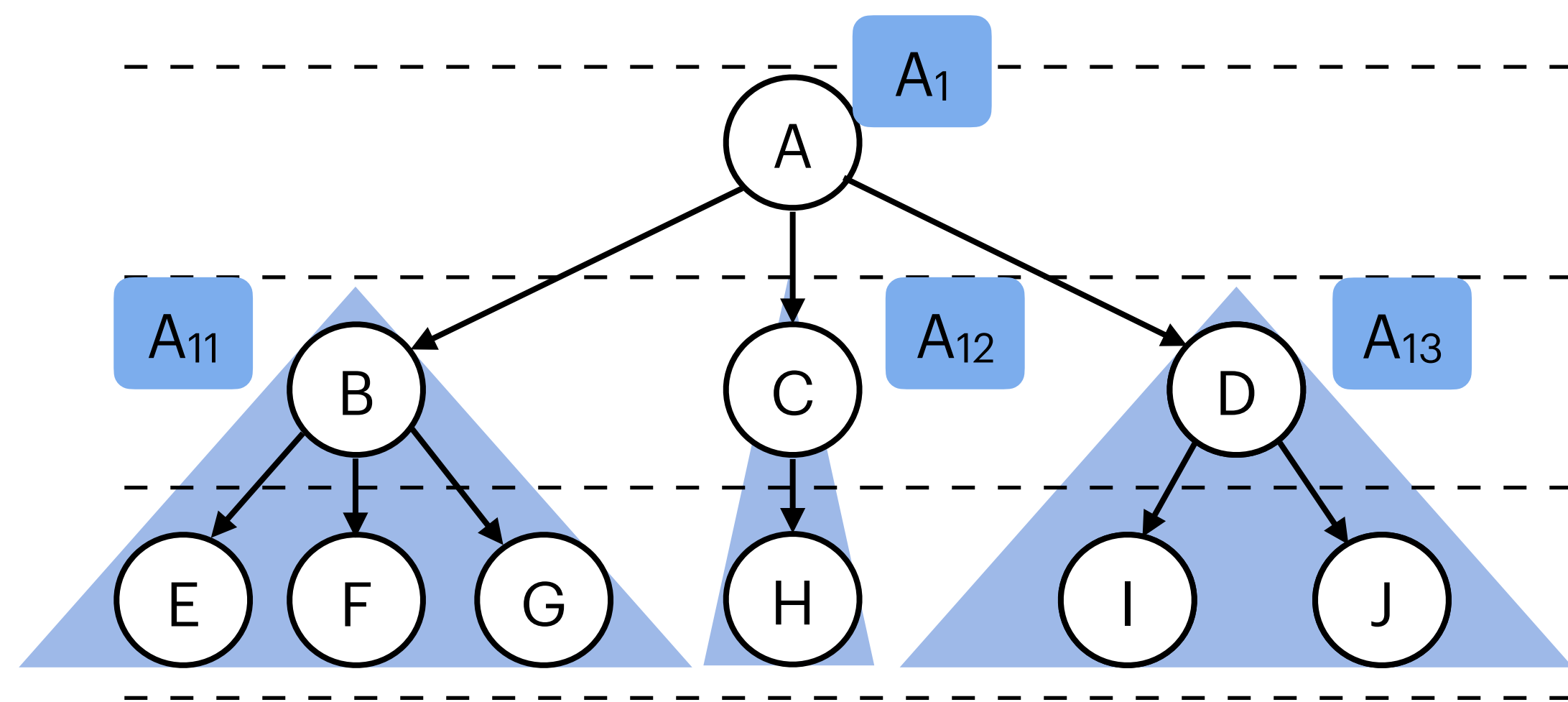
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



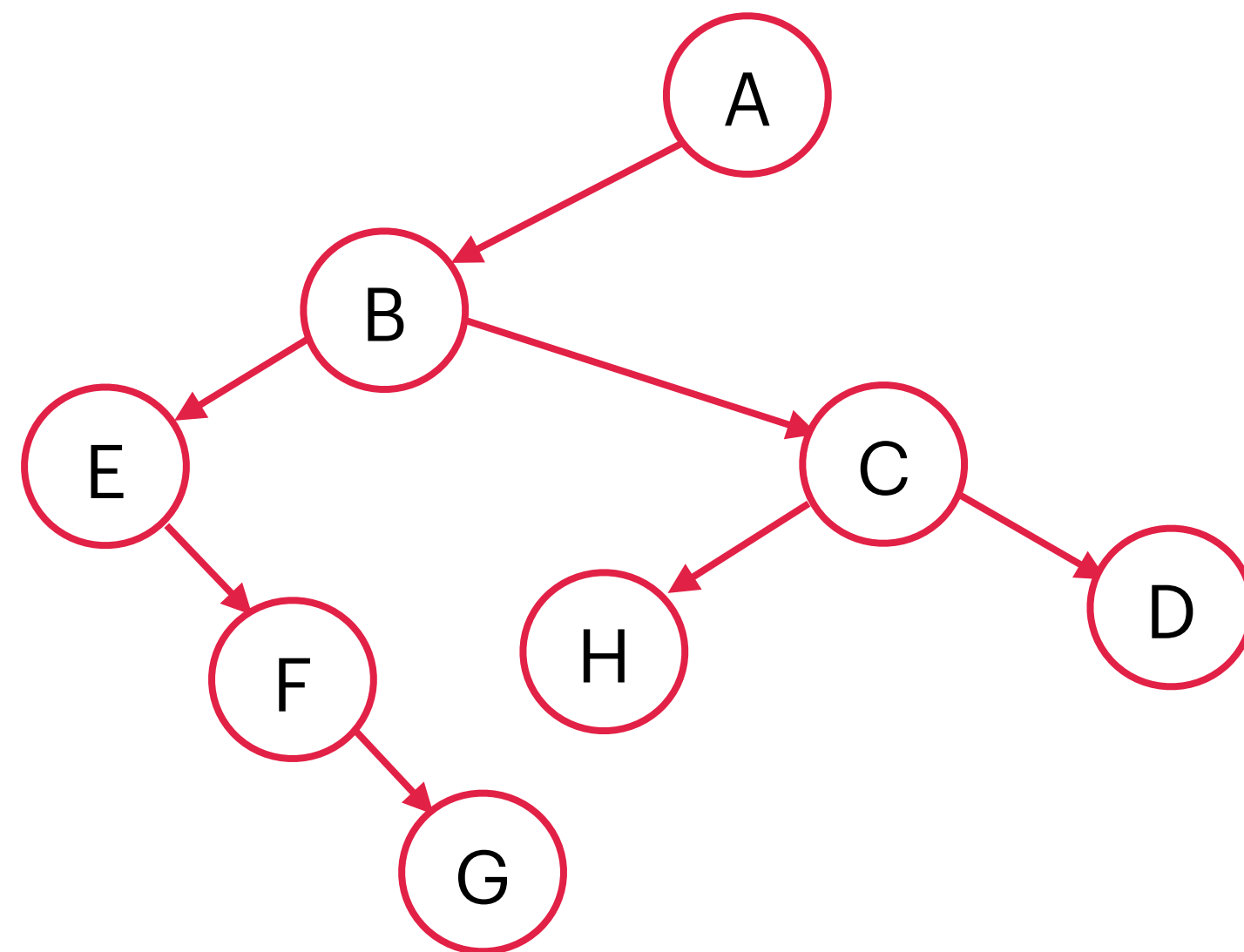
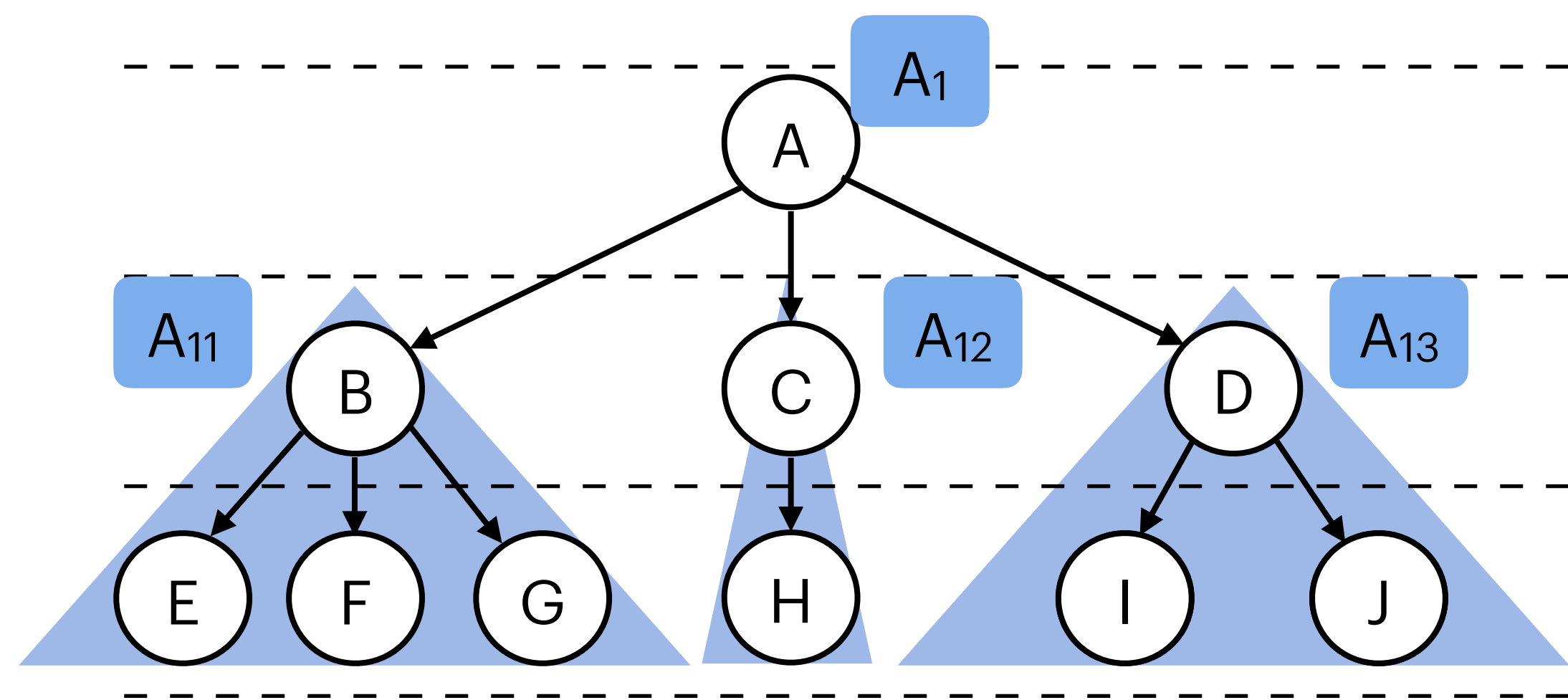
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



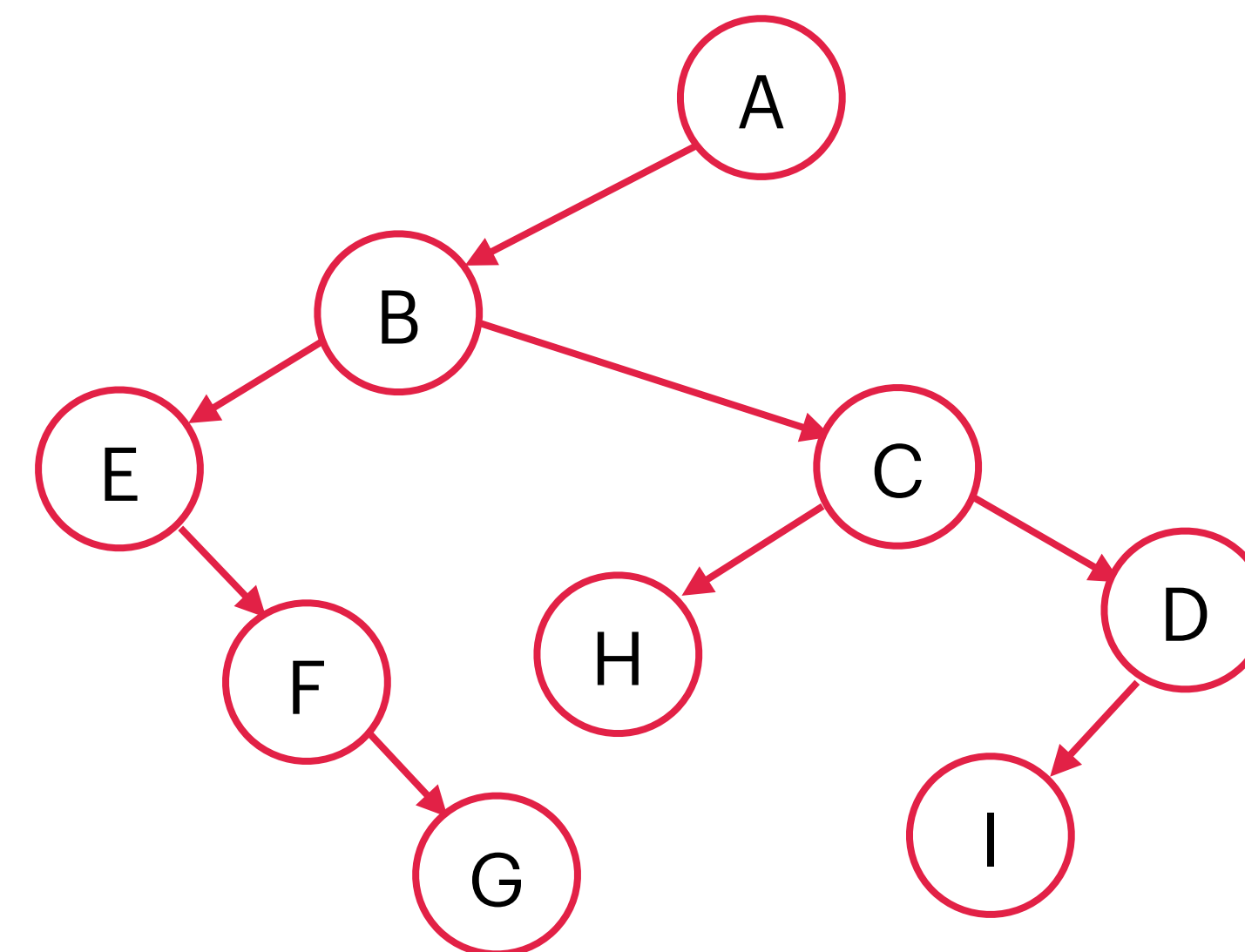
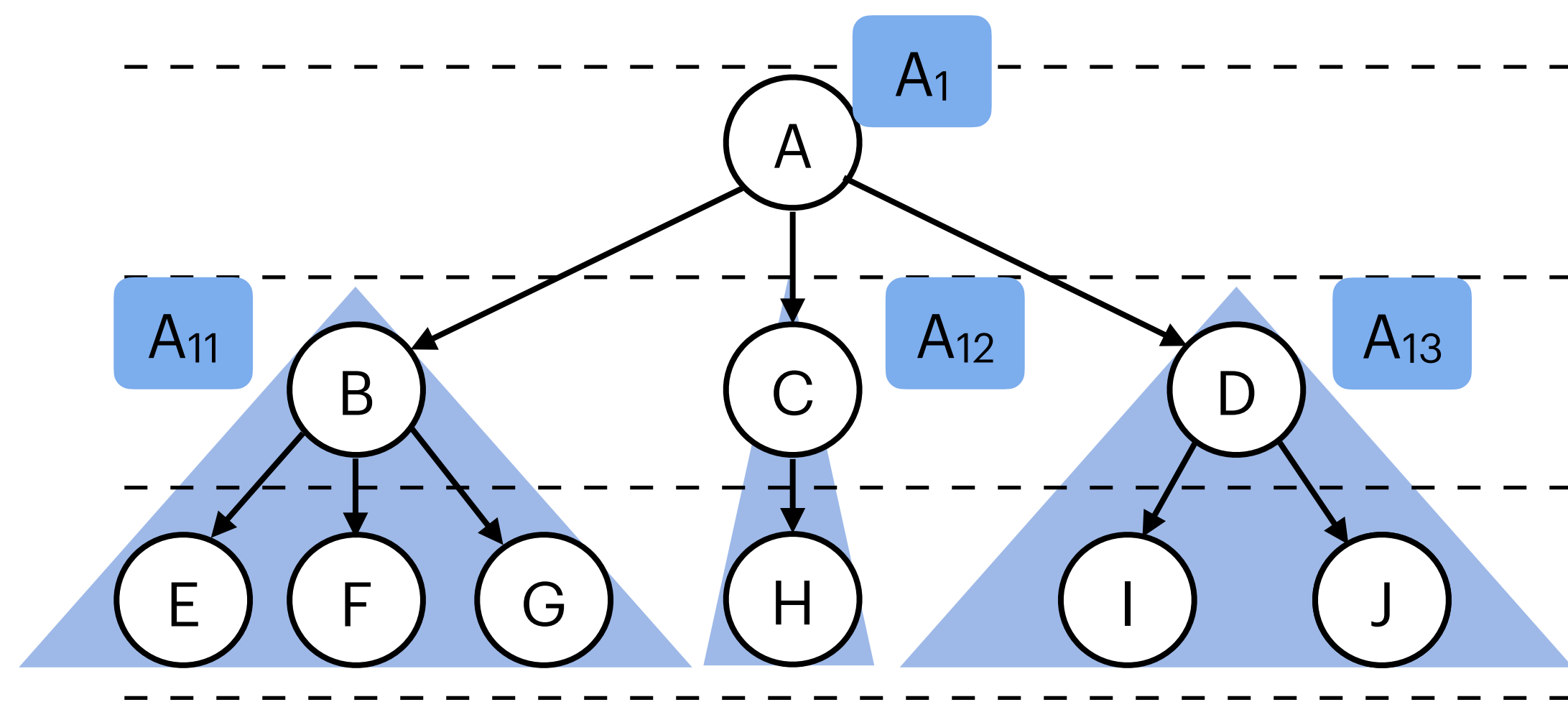
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin



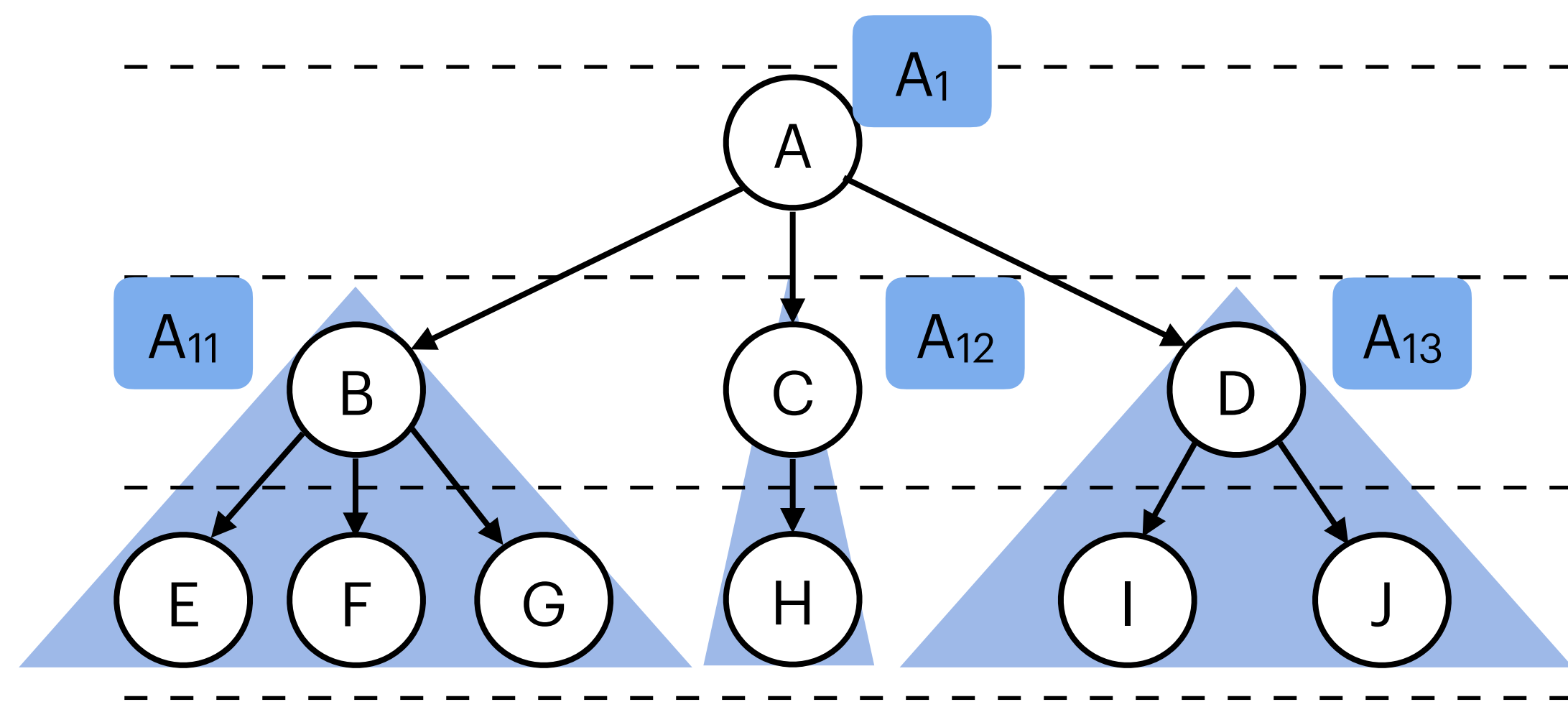
	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin

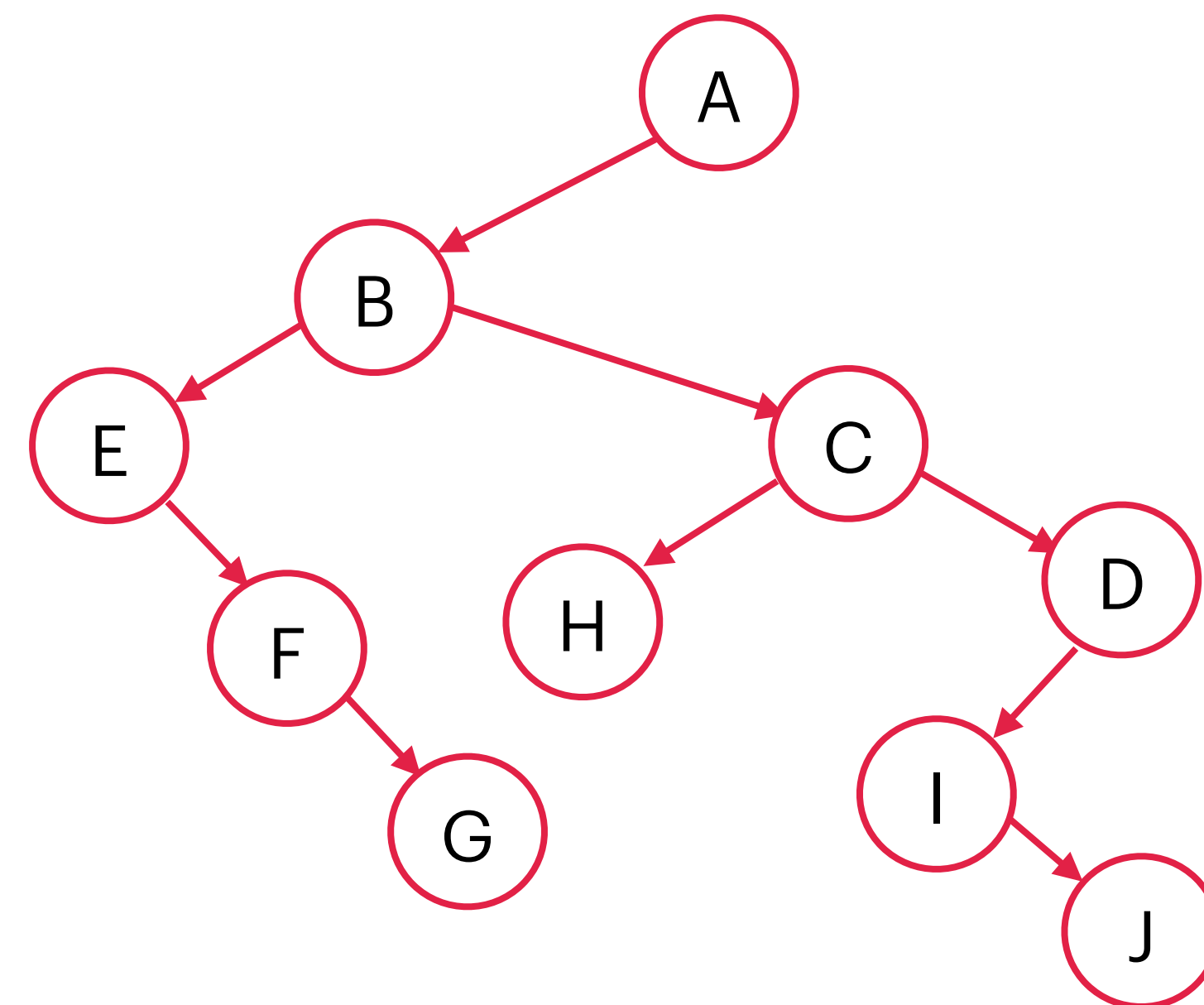


	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0

Transformarea AGen în ABin

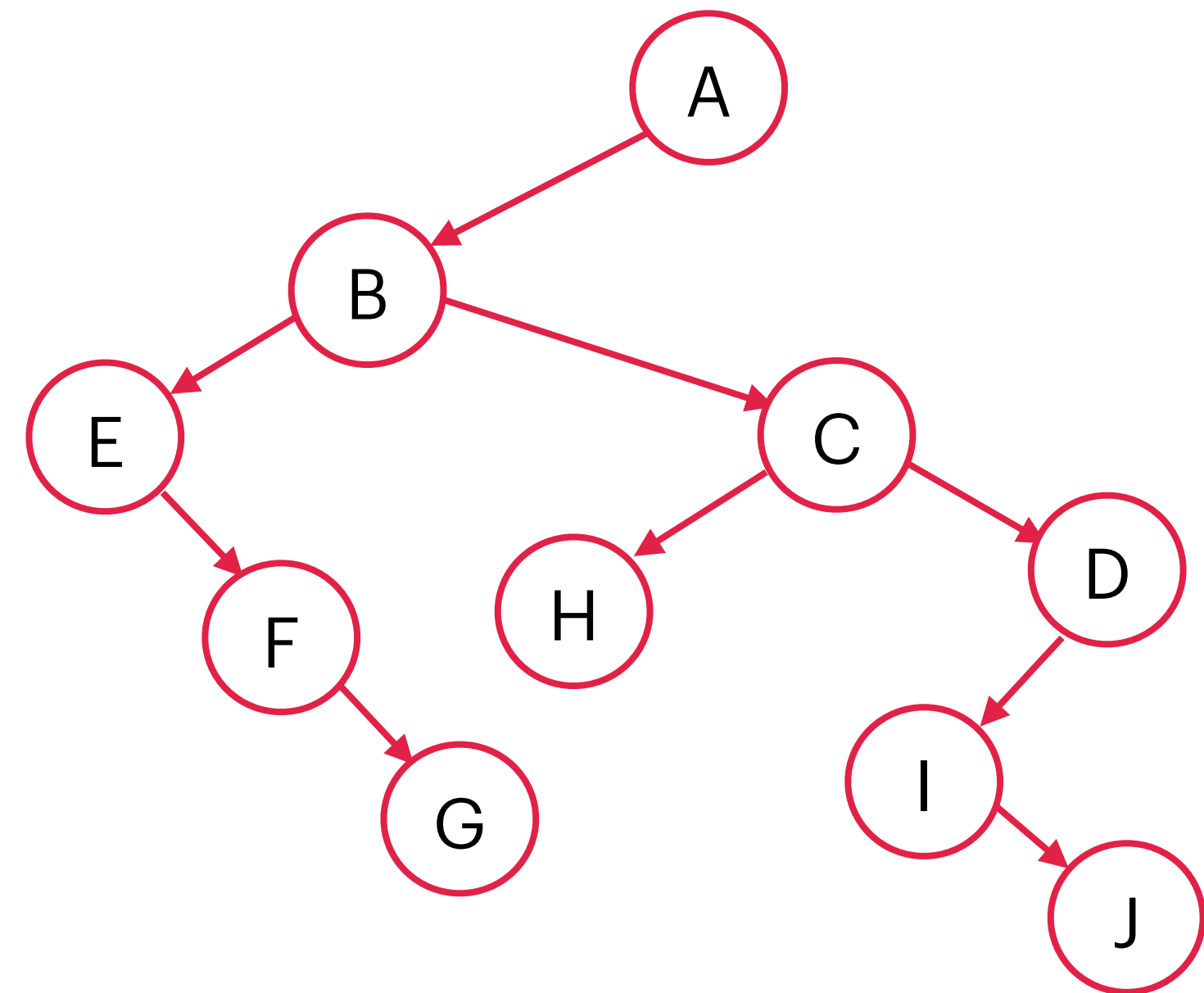


	0	1	2	3	4	5	6	7	8	9	10
Val	_	A	B	C	D	E	F	G	H	I	J
PF	0	2	5	8	9	0	0	0	0	0	0
FD	0	0	3	4	0	6	7	0	0	10	0



Implementare

Implementare bazată pe **tablouri** (utilizată mai rar)



	0	1	2	3	4	5	6	7	8	9	10
Data	_	A	B	C	D	E	F	G	H	I	J
St	0	2	5	8	9	0	0	0	0	0	0
Dr	0	0	3	4	0	6	7	0	0	10	0

Implementare

Un alt mod de reprezentare bazat pe tablouri se bazează pe:

Lema 1: Numărul maxim de noduri al nivelului i al unui arbore binar este 2^{i-1}

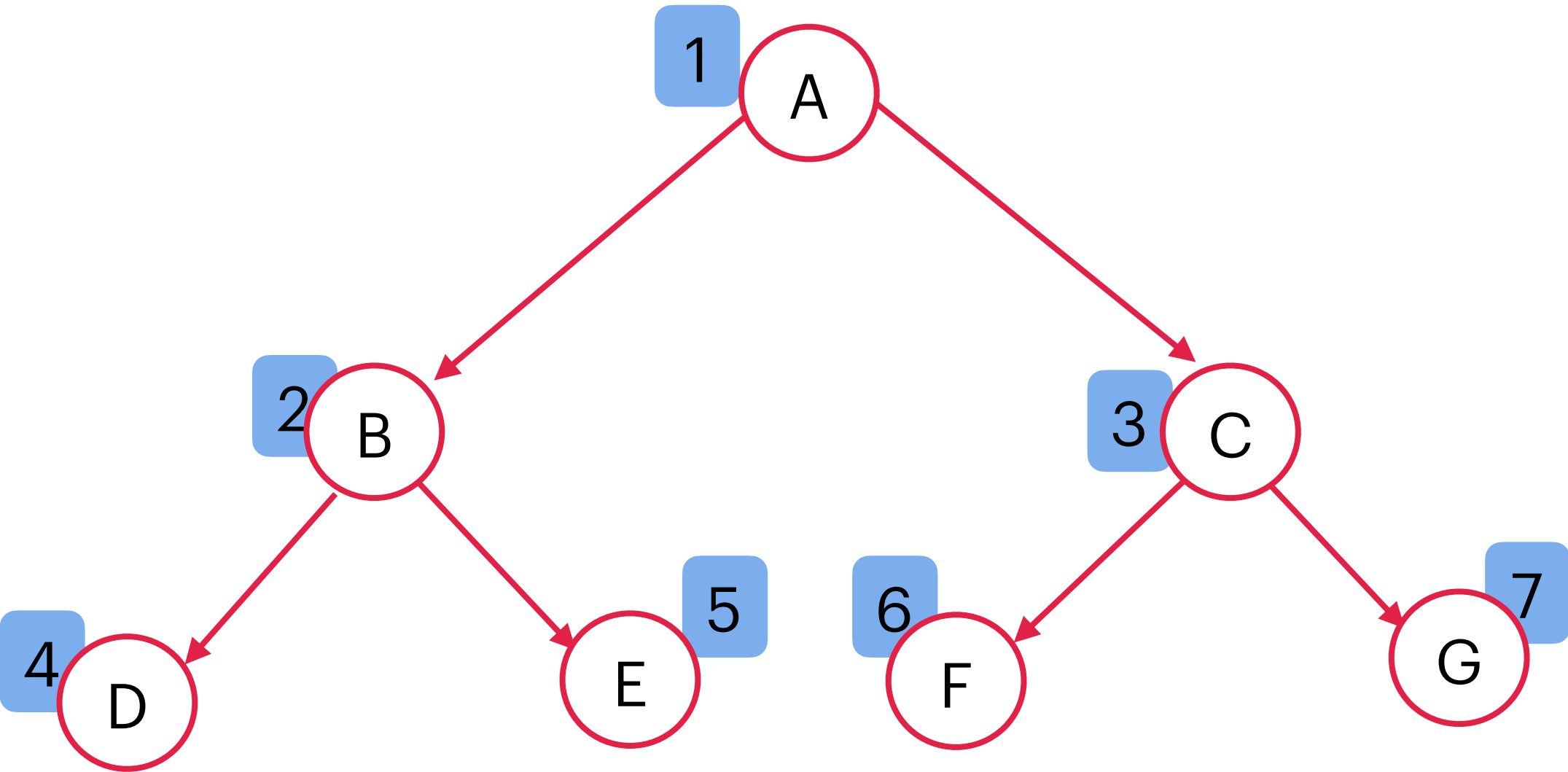
Deci, numărul maxim de noduri al unui arbore binar de înălțime h

$$NrMax = \sum_{i=1}^h 2^{i-1} - 1; h > 0$$

Def: Arborele binar de înălțime h care are exact 2^h-1 noduri se numește **arbore binar plin de înălțime h**

Se numerotează secvențial nodurile unui arbore binar plin de înălțime h , începând cu rădăcina și continuând de sus în jos și de la stânga la dreapta

Asociem numărul fiecărui nod cu poziția într-un tablou liniar



	1	2	3	4	5	6	7
Val	A	B	C	D	E	F	G

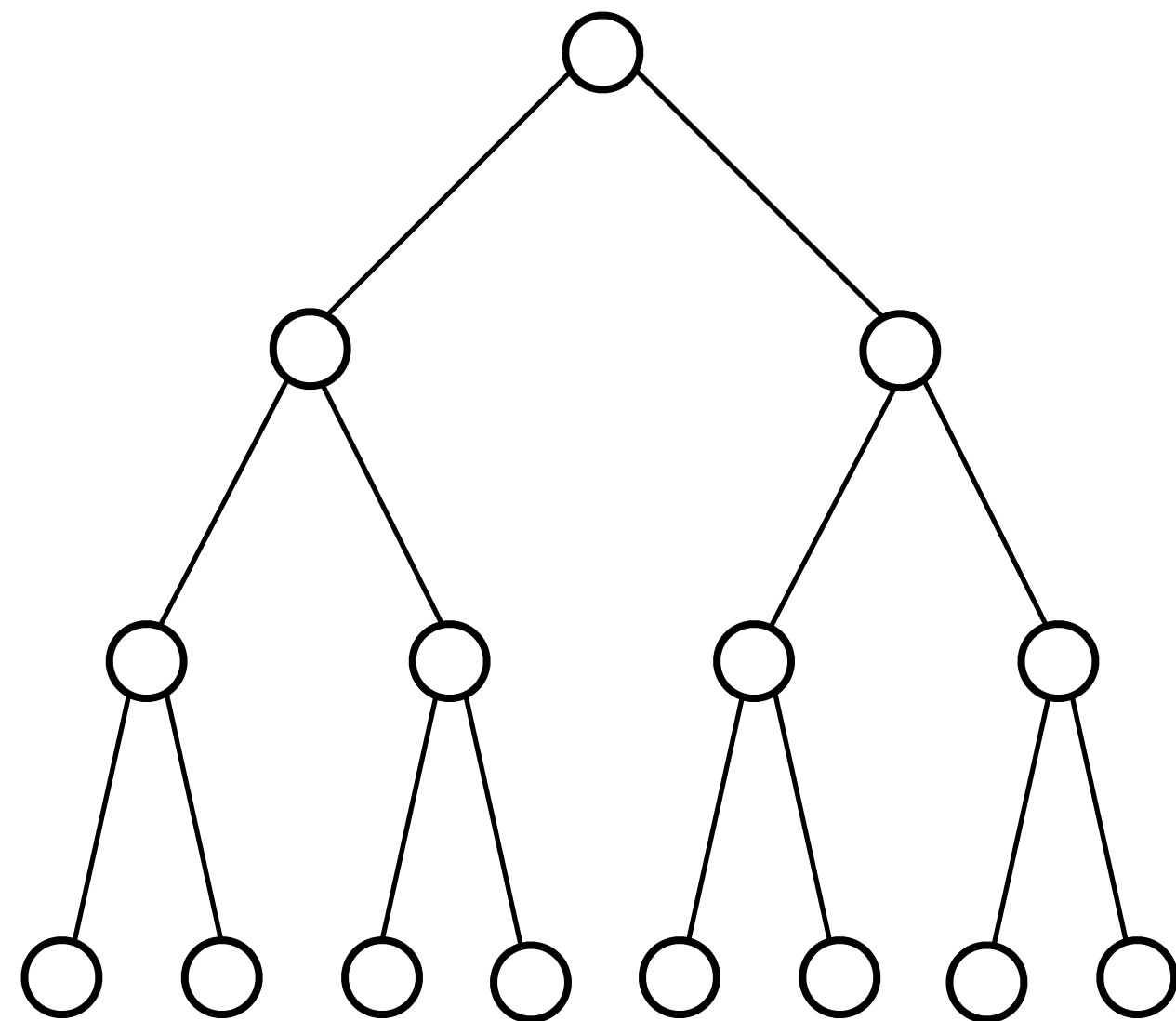
Implementare

Def: Un arbore binar cu n noduri de înălțime h se numește arbore **binar complet** dacă nodurile sale corespund nodurilor care sunt numerotate de la 1 la n într-un arbore binar plin de înălțime h .

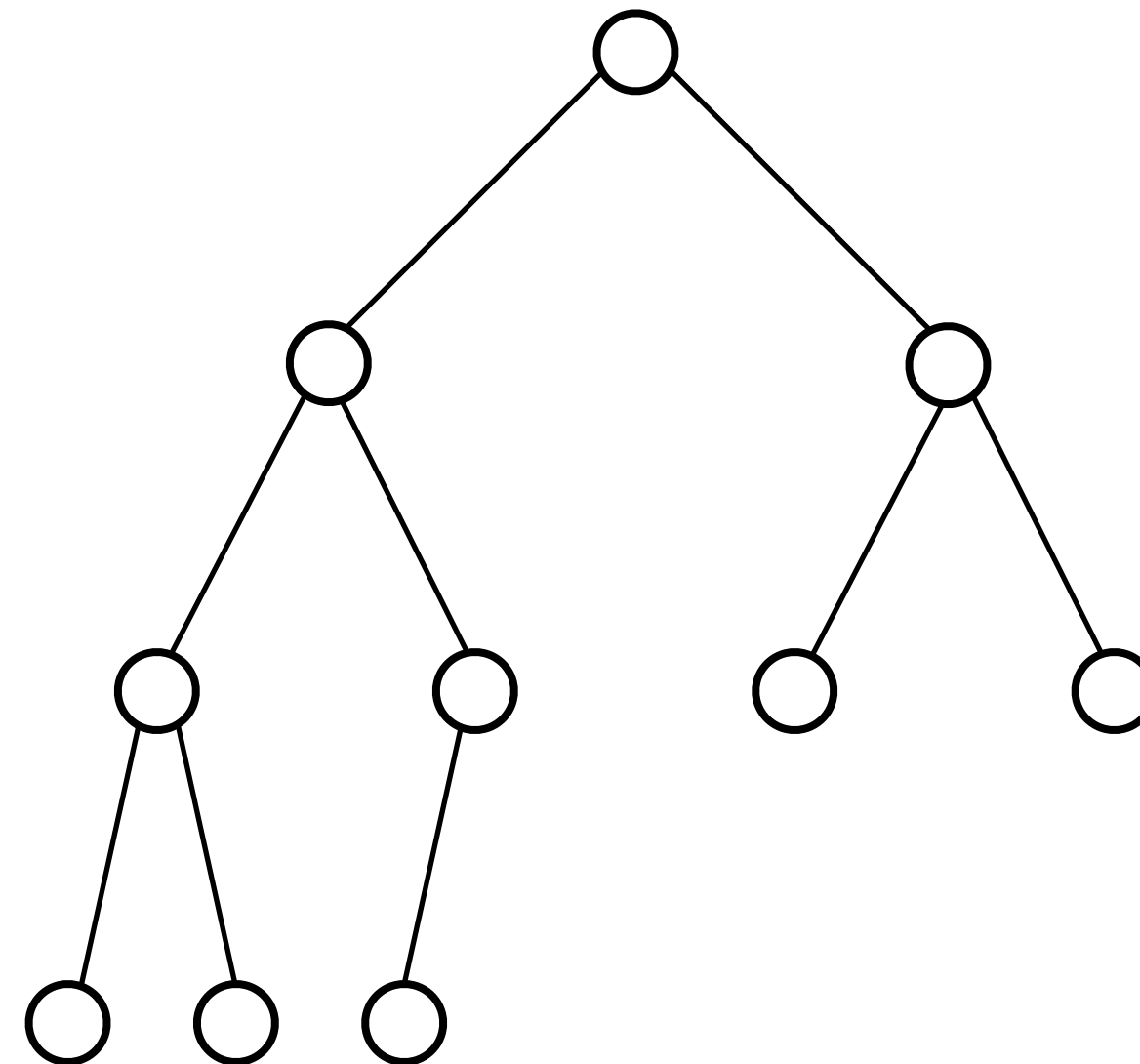
O consecință a definiției este că într-un **arbore binar complet** nodurile terminale apar pe **cel mult două niveluri adiacente**

Diferența între un arbore binar complet și un arbore binar plin este, ambii de înălțime h , este că primul are un număr n de noduri $n < 2^h - 1$ iar cel de-al doilea are exact $2^h - 1$ noduri

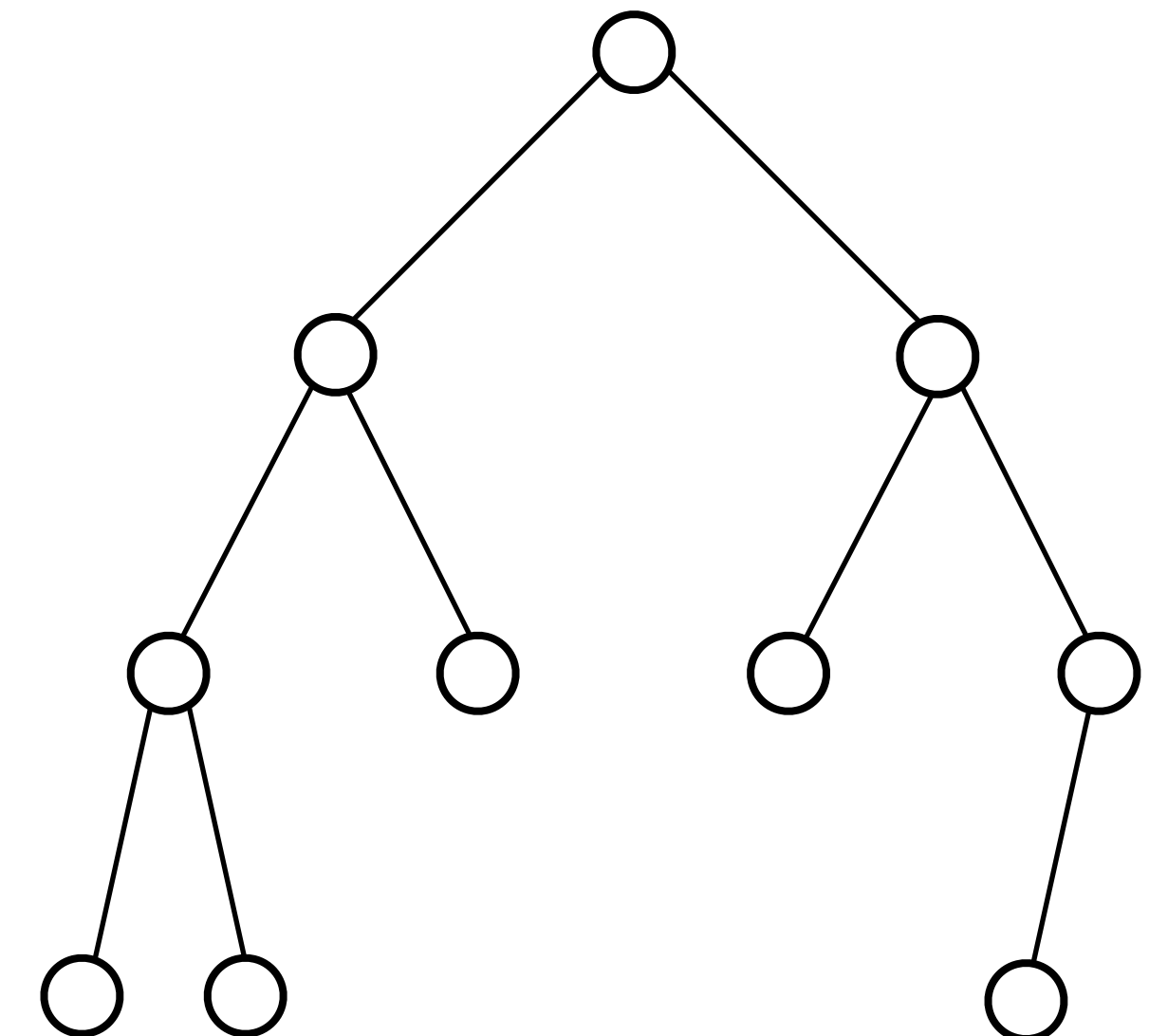
Ambii arbori sunt arbori binari de înălțime minimă, care au proprietatea ca nodurile terminale sunt distribuite pe cel mult două niveluri ale arborelui



Arbore binar plin



Arbore binar complet



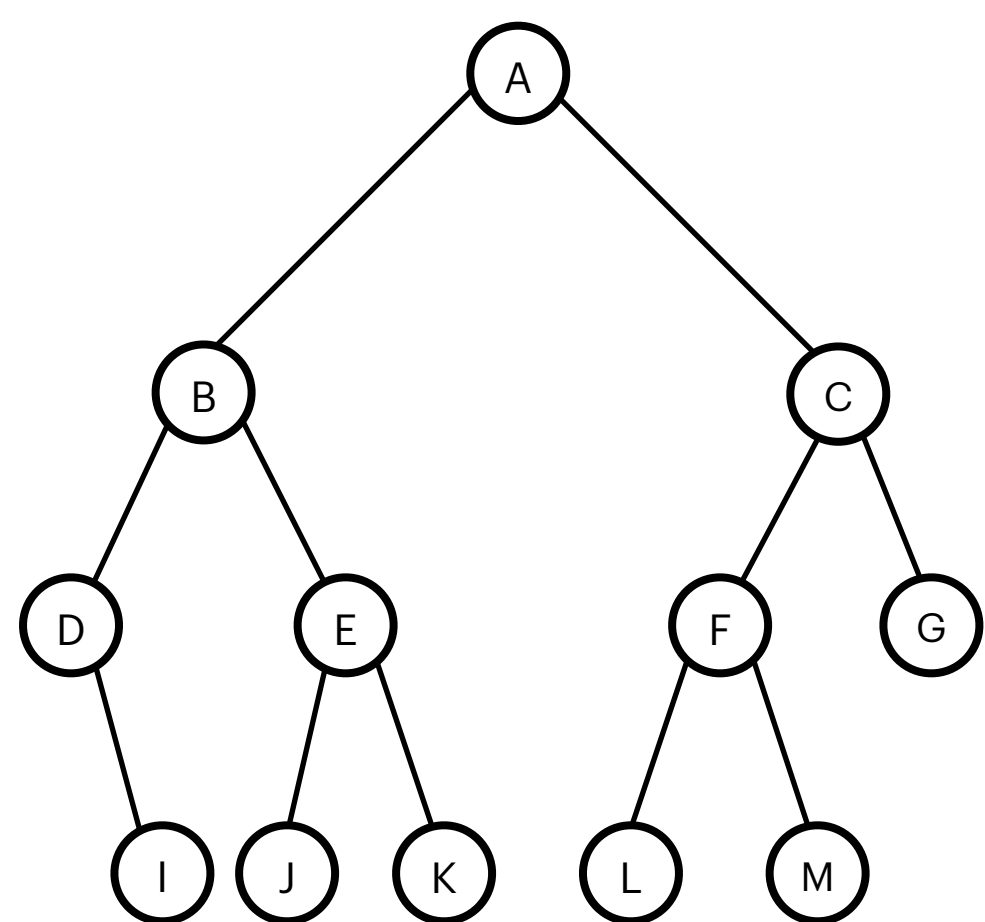
Arbore binar de înălțime minimă (echilibrat)

Implementare

Lema2: Dacă se reprezintă un **arbore binar complet** conform cu cele precizate anterior, atunci pentru orice nod având indicele i , $1 \leq i \leq n$ sunt valabile relațiile:

- (1) Parinte(i) este nodul cu indicele $i/2$ dacă $i \neq 1$. Dacă $i = 1$ nodul indicat este cel rădăcină și nu are părinte
- (2) FiuStang(i) este nodul având indicele $2 * i$ dacă $2 * i \leq n$. Dacă $2 * i > n$, atunci nodul i nu are fiu stâng
- (3) FiuDrept(i) este nodul având indicele $2 * i + 1$ dacă $2 * i + 1 \leq n$. Dacă $2 * i + 1 > n$, atunci nodul i nu are fiu drept.

Acest mod de implementare poate fi folosit pentru orice structură de arbore binar, dar, în general este o metodă ineficientă. Nodurile lipsă pentru ca arborele să devină complet se completează cu nodul vid.



Avantaje: simplitate, abseța legăturilor, parcurgerea simplă în ambele sensuri

Dezavantaje: dimensiune limitată, inserții / suprimări relativ complicate, ineficient în cazul arborilor rari

Se recomandă: pentru arbori binari cu o dinamică redusă a modificărilor, cu multe parcurgeri sau căutări

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
\0	A	B	C	D	E	F	G	\0	I	J	K	L	M	\0	\0

I m p l e m e n t a r e

Implementare cu ajutorul pointerilor

Cea mai **flexibilă** metodă de implementare a arborilor binari

Înlătură **limitările** reprezentării secvențiale bazate pe structura tablou

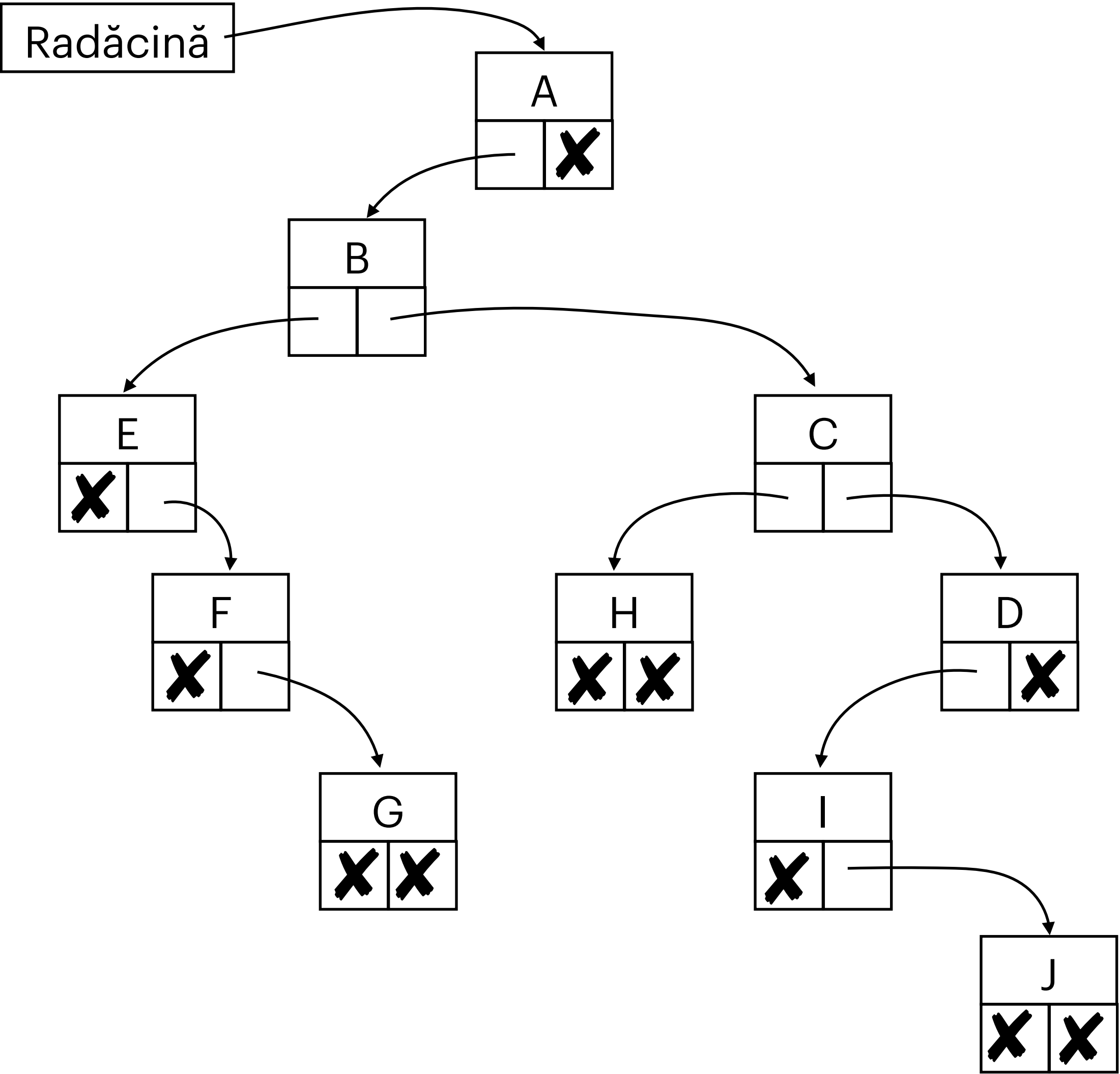
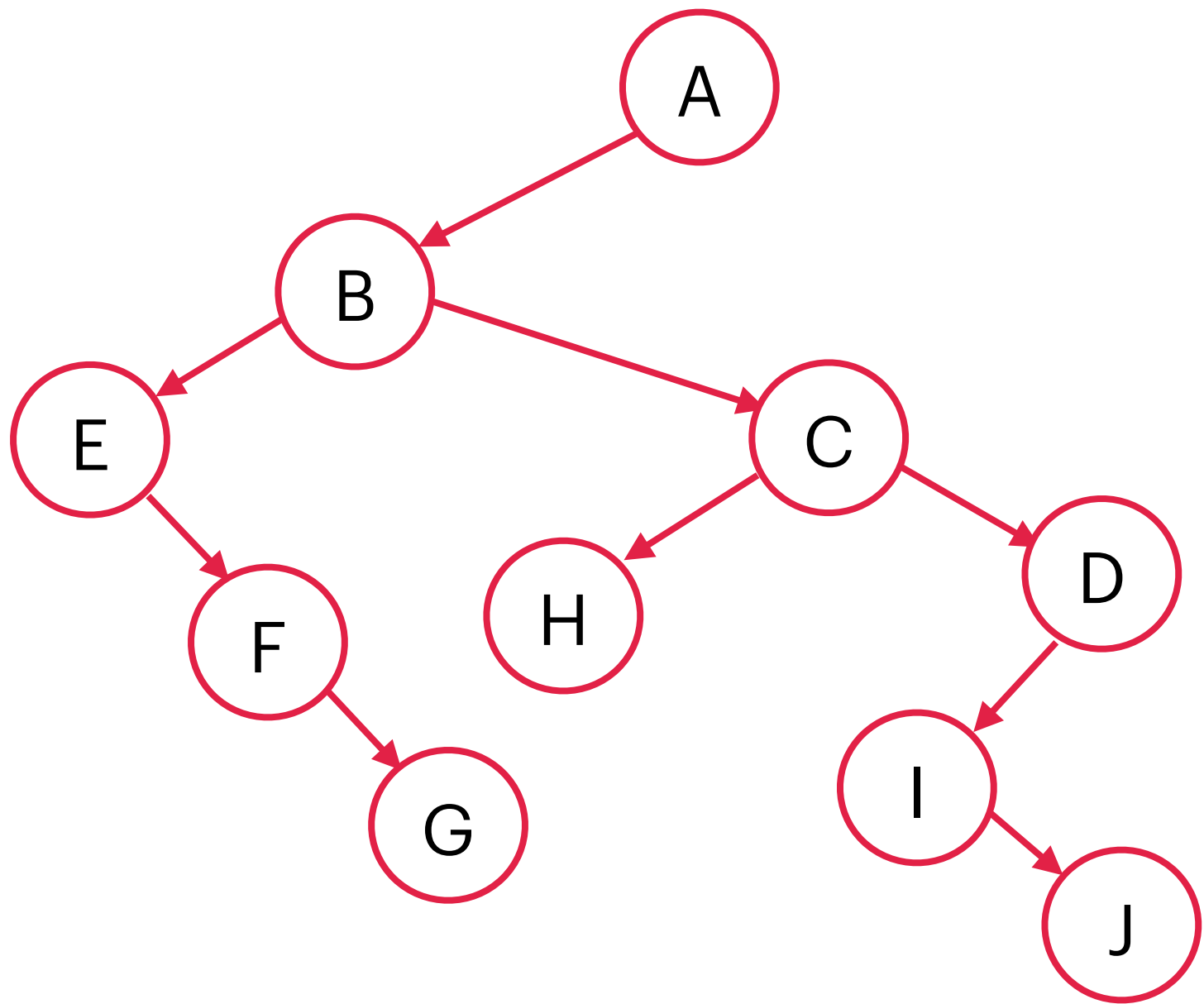
Face uz de proprietățile **recursive** ale structurii arbore, implementarea realizându-se cu **structuri de date dinamice**

```
typedef struct tip_nod{  
    TKey key;  
    struct TNod * st;  
    struct TNod * dr;  
} TNod
```

```
typedef TNod * RefNod
```

Implementare

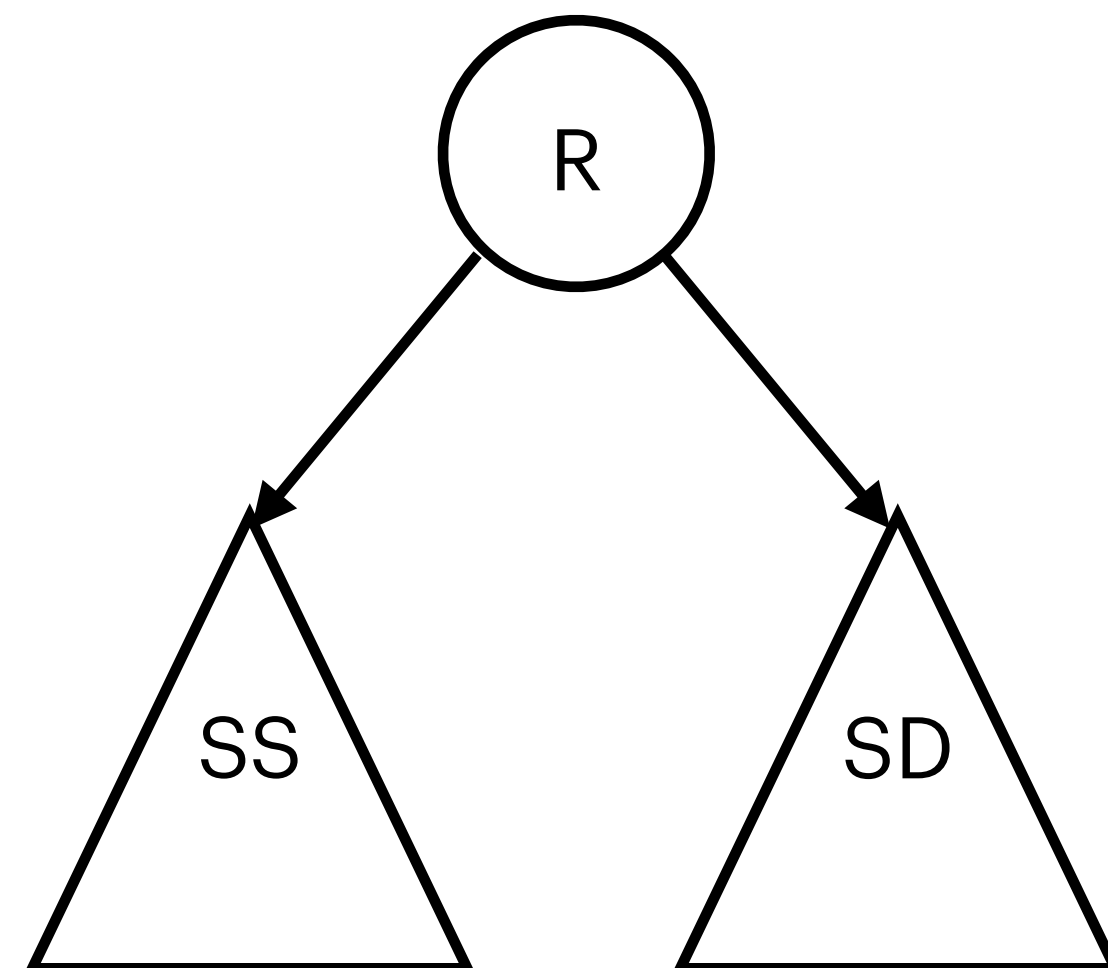
Implementare cu ajutorul pointerilor



T r a v e r s a r e

Traversarea arborilor binari derivă direct prin particularizare din tehnicile de traversare a arborilor generalizați

Traversarea bazată pe căutarea în adâncime



Preordine: R, Preordine(SS), Preordine(SD)

Inordine: Inordine(SS), R, Inordine(SD)

Postordine: Postordine(SS), Postordine(SD), R

Arbori Binari cu Legături

Threaded Trees

Implementarea cu pointeri permit parcurgerea simplă în manieră **descendentă** dar **parcurgerea ascendentă** e relativ dificilă

Se poate memora în fiecare moment al parcurgerii, pe lângă nodul curent și părintele acestuia

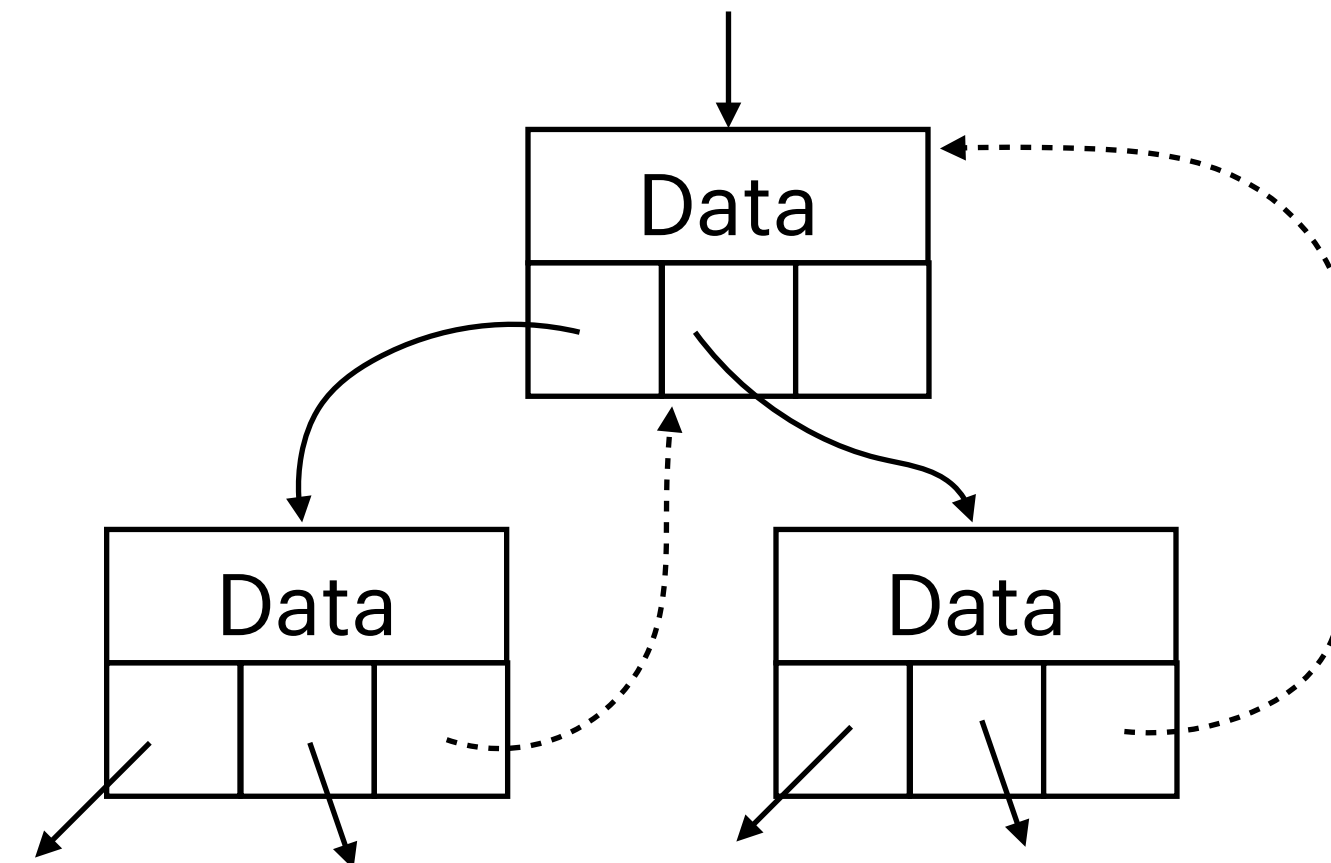
Dezavantaj: consum mai ridicat de memorie

Avantaj: grad de complexitate redus al algoritmului de parcurgere

Alternativ se poate memora în structura nodului o legătură către părinte. Dacă nodul e suficient de complex, un pointer în plus e nesemnificativ

```
typedef struct tip_nod{
    TKey key;
    struct TNod * st;
    struct TNod * dr;
    struct TNod * parinte;
} TNod;

typedef TNod * RefNod
```



Arbori Binari cu Legături

O altă posibilitate de implementare se bazează pe observația că o **mare** parte a **nodurilor** au câmpurile indicator (pointeri) **nule**.

Aceste câmpuri pot fi modificate să indice noduri în arbore care preced sau succed nodul curent într-o anumită ordine de traversare

Referindu-ne la traversarea în inordine, putem conveni că în fiecare nod

În locul înlănțuirii vide spre dreapta apare legătura la nodul următor în parcurgerea în inordine

În locul înlănțuirii vide spre stânga apare legătura la nodul anterior în parcurgerea în inordine

Întrucât câmpurile indicator au fost supraîncărcate (overloaded) trebuie făcute modificări care să evidențieze dacă e vorba de înlănțuiri sau legături

Înlănțuirile indică fi nodului curent iar legăturile indică predecesorii sau succesorii în parcurgerea în inordine

O astfel de structură se numește arbore cu legături (threaded trees)

Un arbore cu legături poate fi:

Parțial - dacă numai unul din câmpurile unui nod e folosit ca legătură

Plin - (fully threaded tree) dacă se utilizează ambele câmpuri

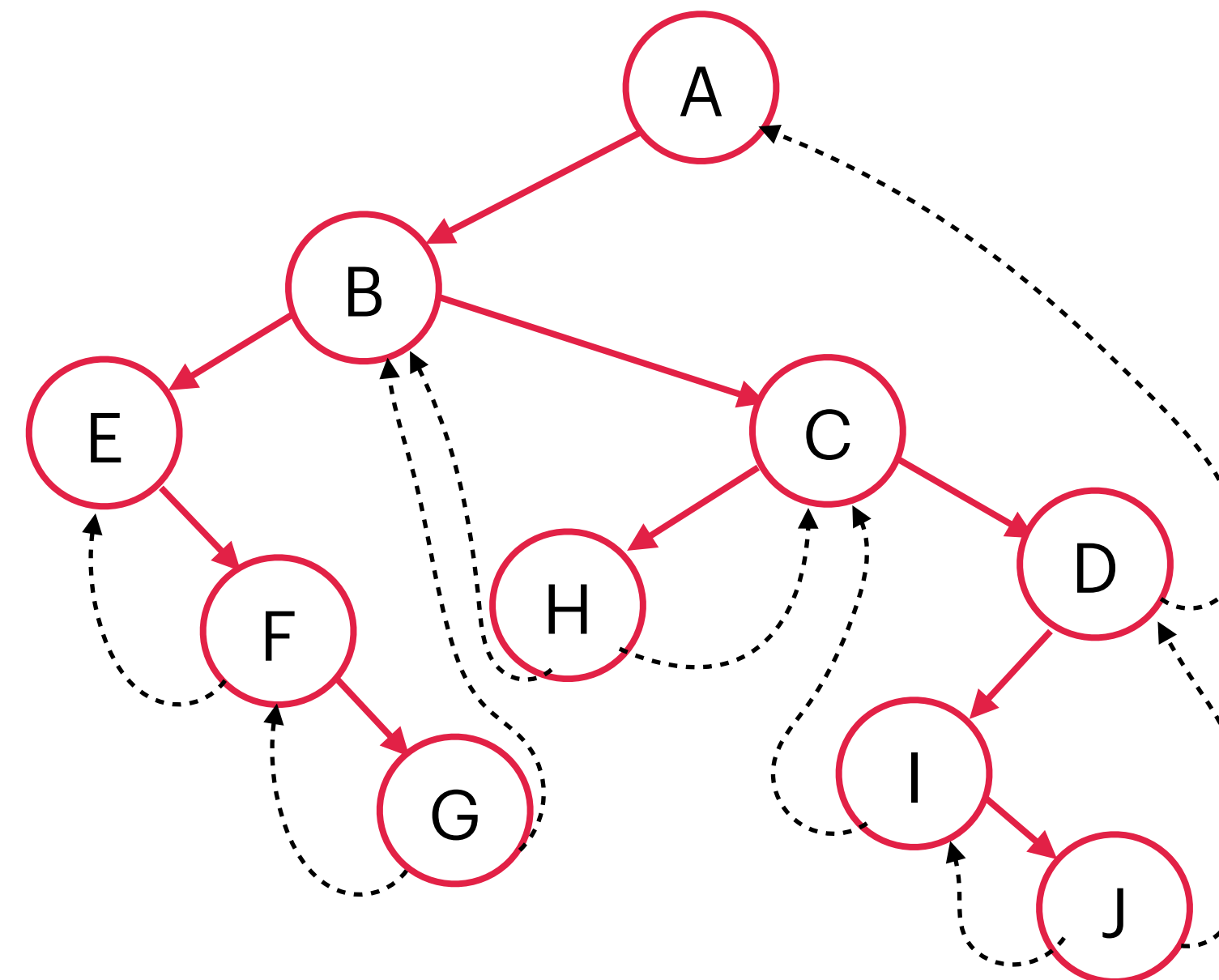
Arbori Binari cu Legături

O altă posibilitate de implementare se bazează pe observația că o **mare** parte a **nodurilor** au câmpurile indicator (pointeri) **nule**.

Inordine: E F G B G H C I J D A

```
typedef struct tip_nod{
    TKey key;
    bool legatura_st, legatura_dr;
    struct TNod * st;
    struct TNod * dr;
    struct TNod * parinte;
} TNod;

typedef TNod * RefNod
```



Arbori Binari cu Legături

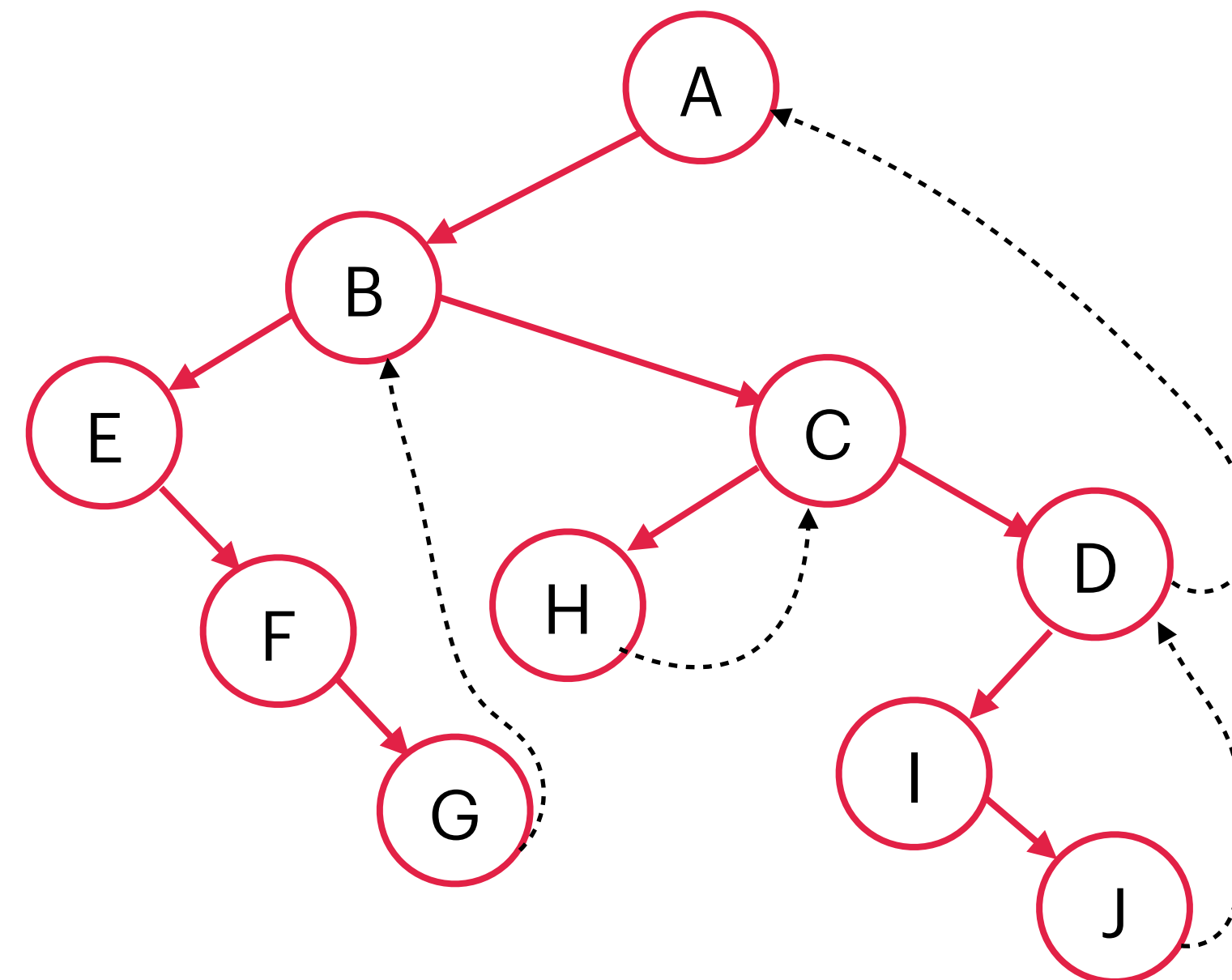
Arbori Binari cu legături parțiale

Se folosesc doar legături spre dreapta

```
typedef struct tip_nod{  
    TKey key;  
    bool legatura_dr;  
    struct TNod * st;  
    struct TNod * dr;  
    struct TNod * parinte;  
} TNod;
```

```
typedef TNod * RefNod
```

Inordine: E F G B G H C I J D A



Arbori Binari Ordonati

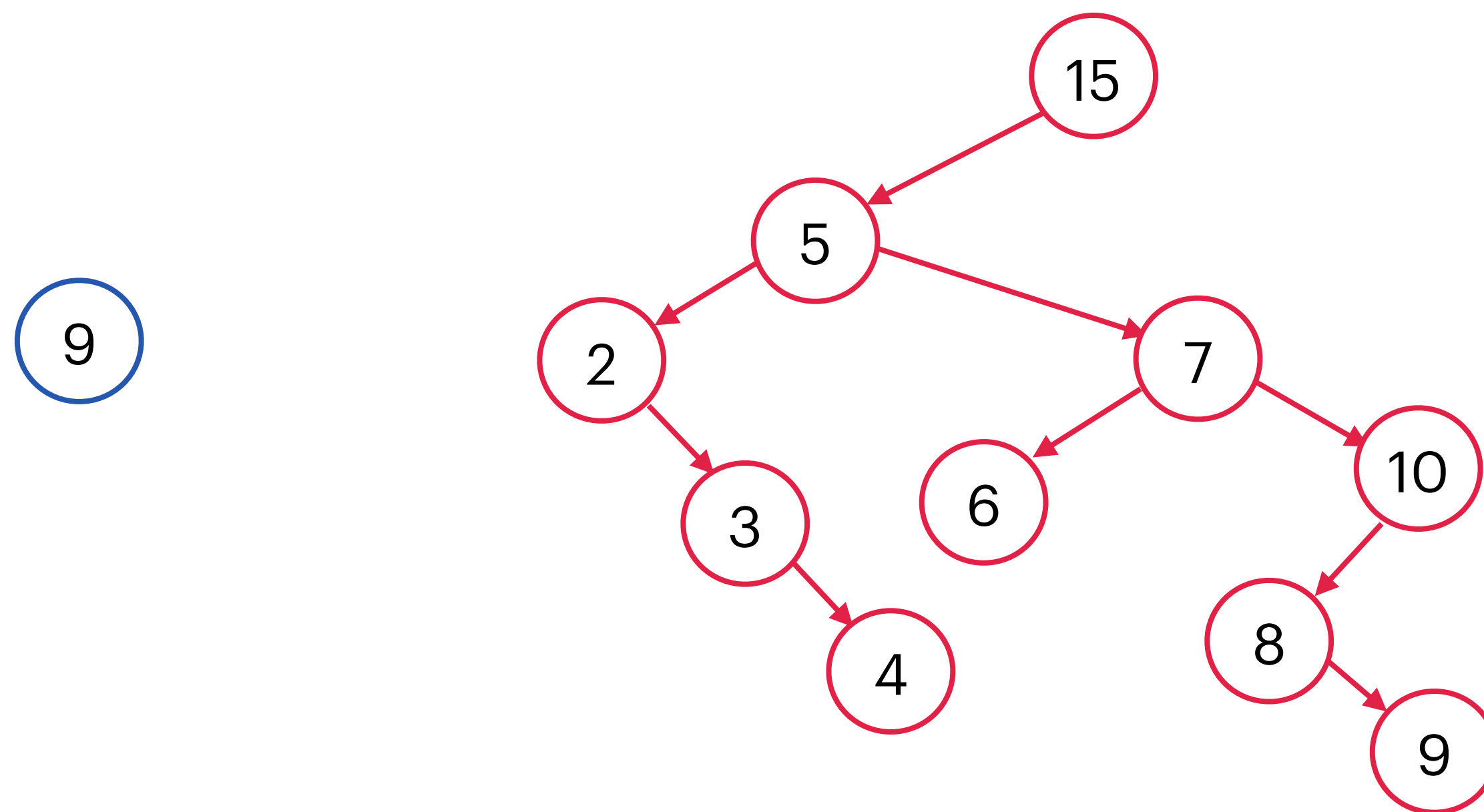
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

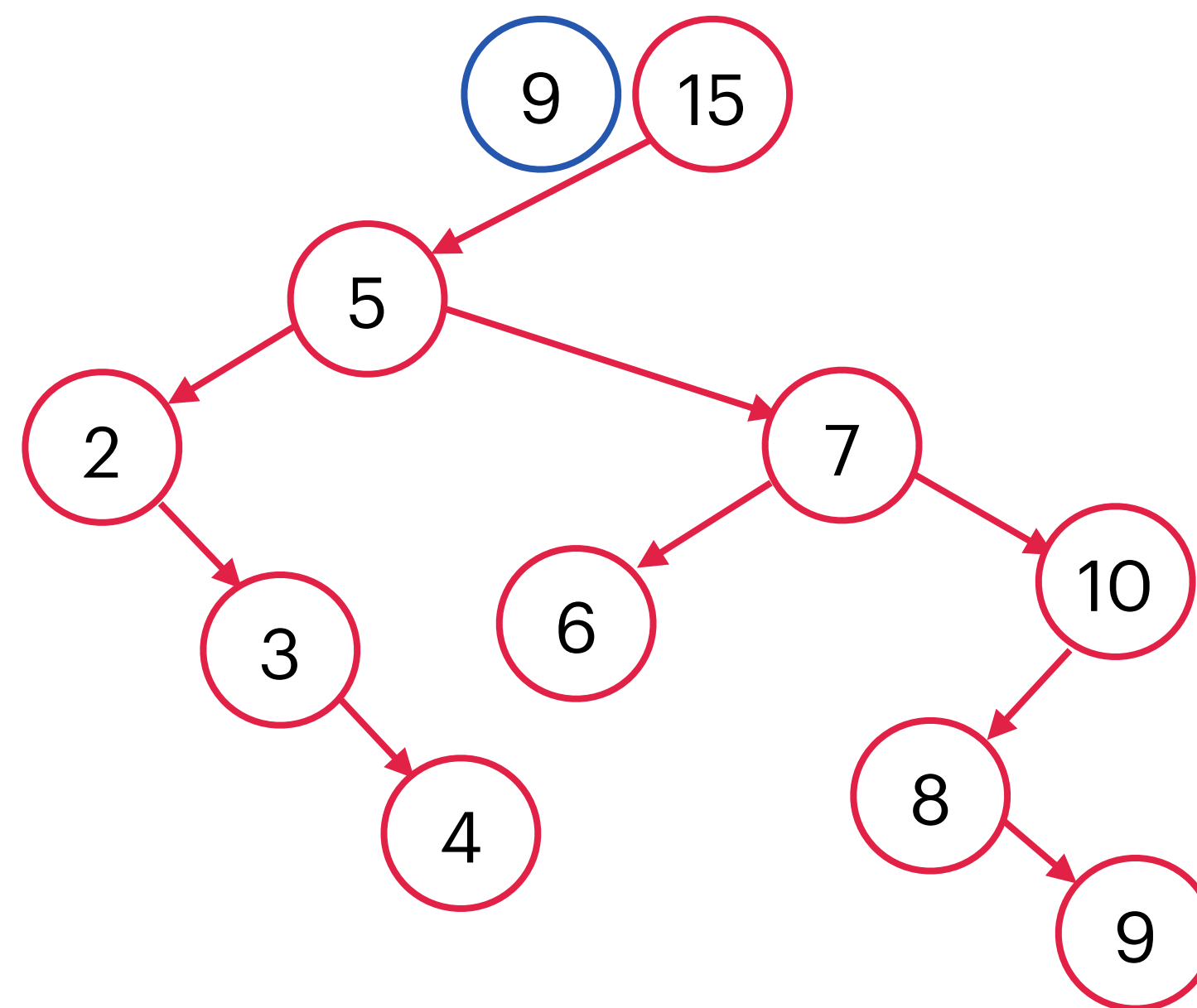
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

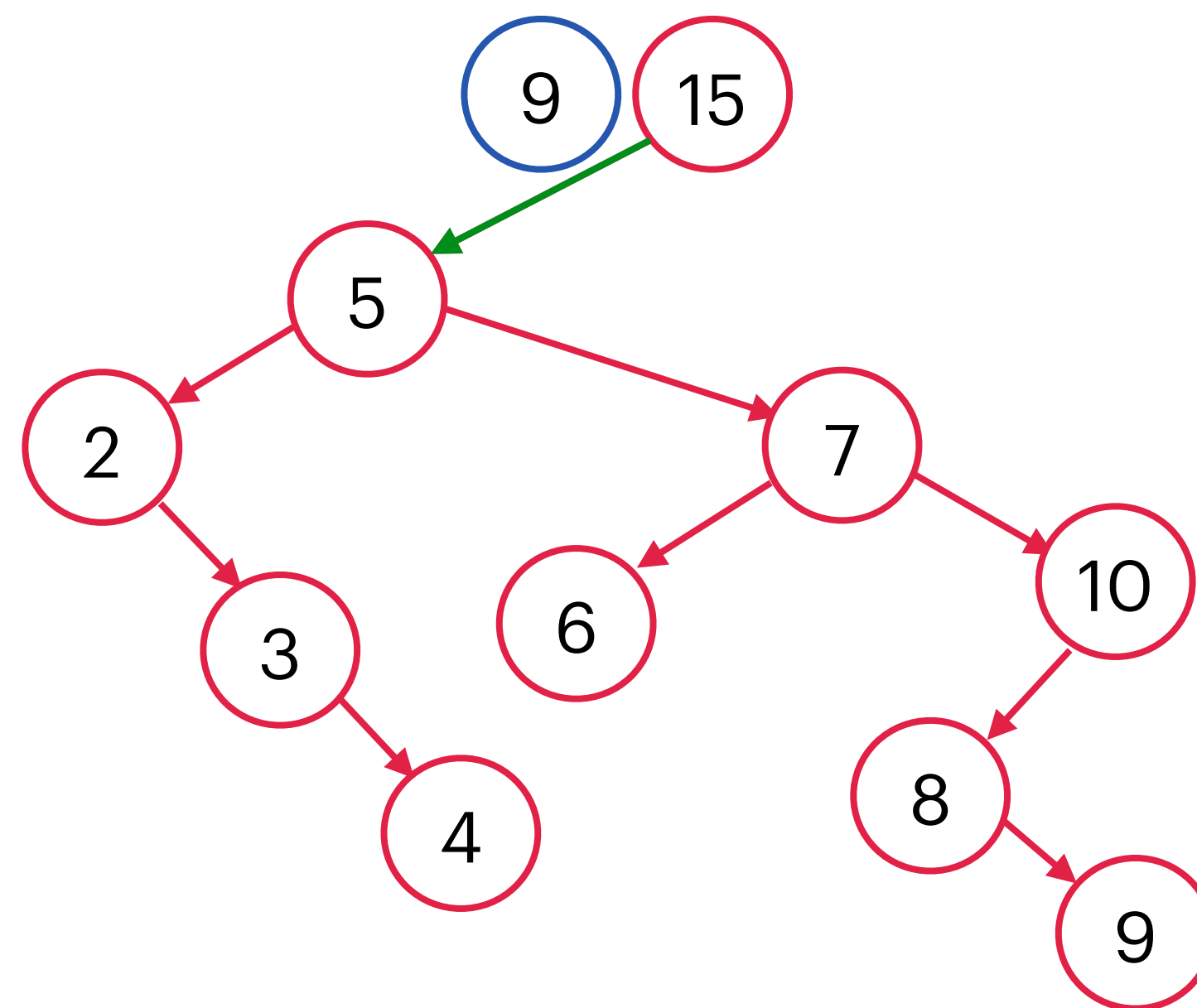
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

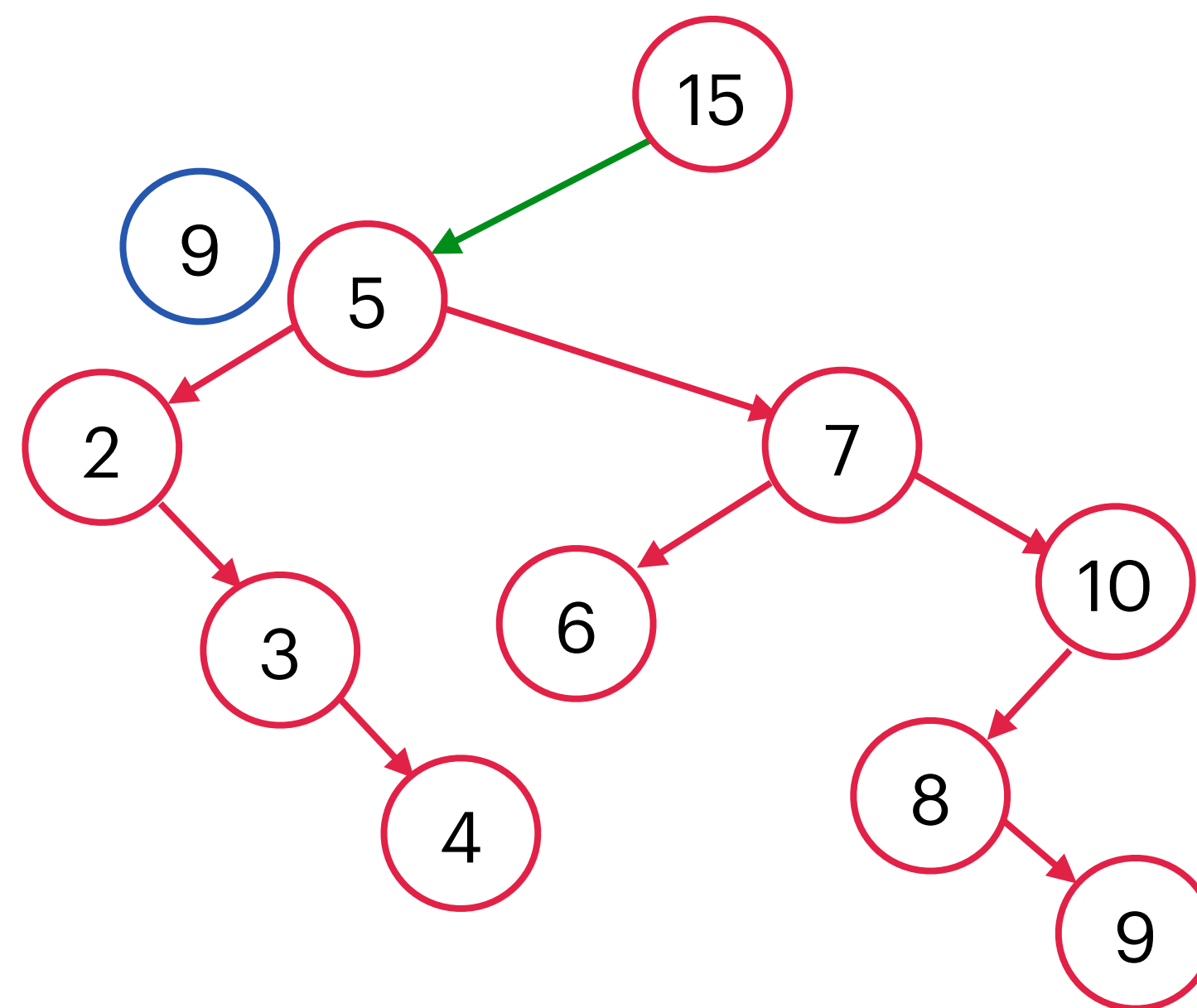
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

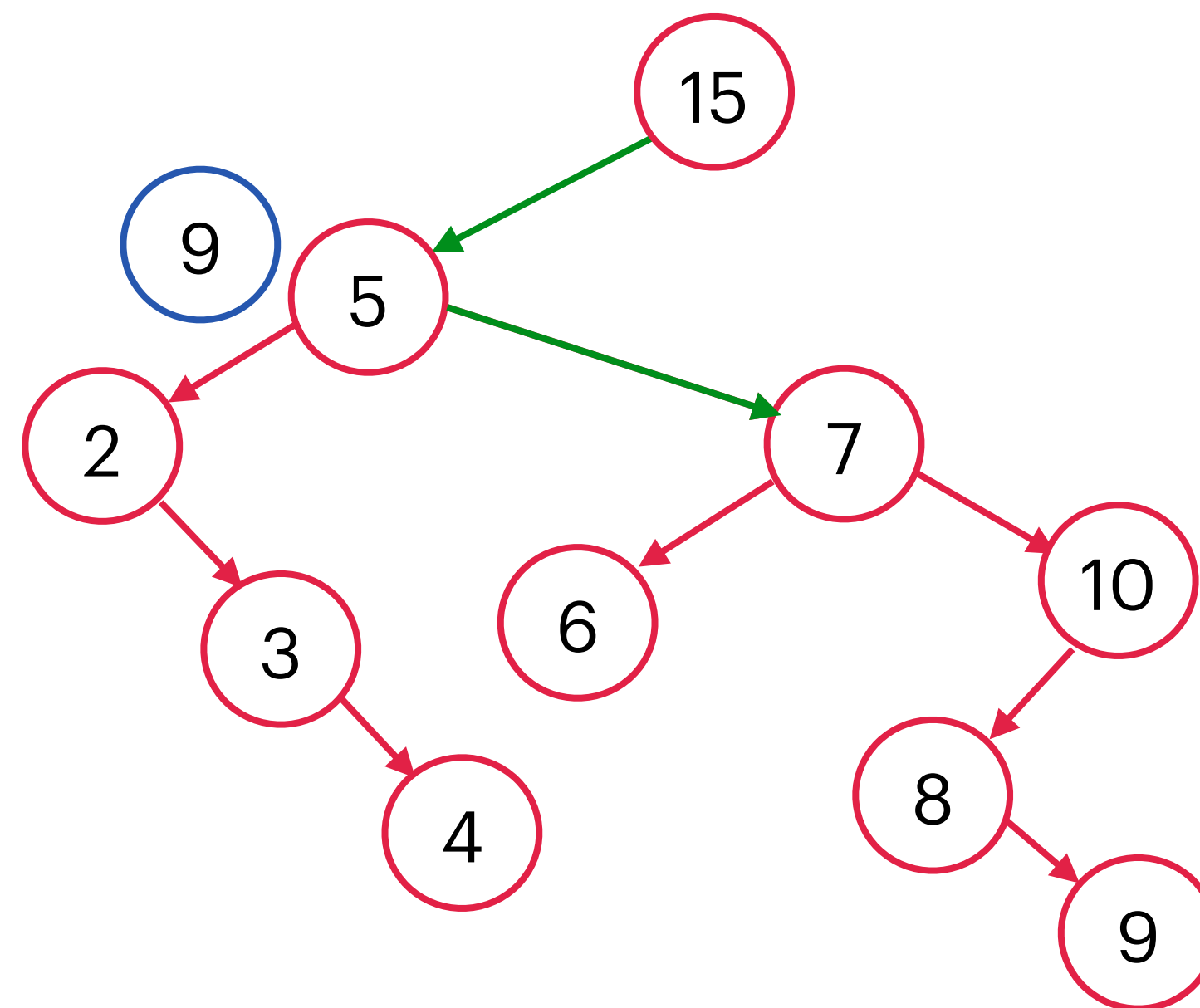
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

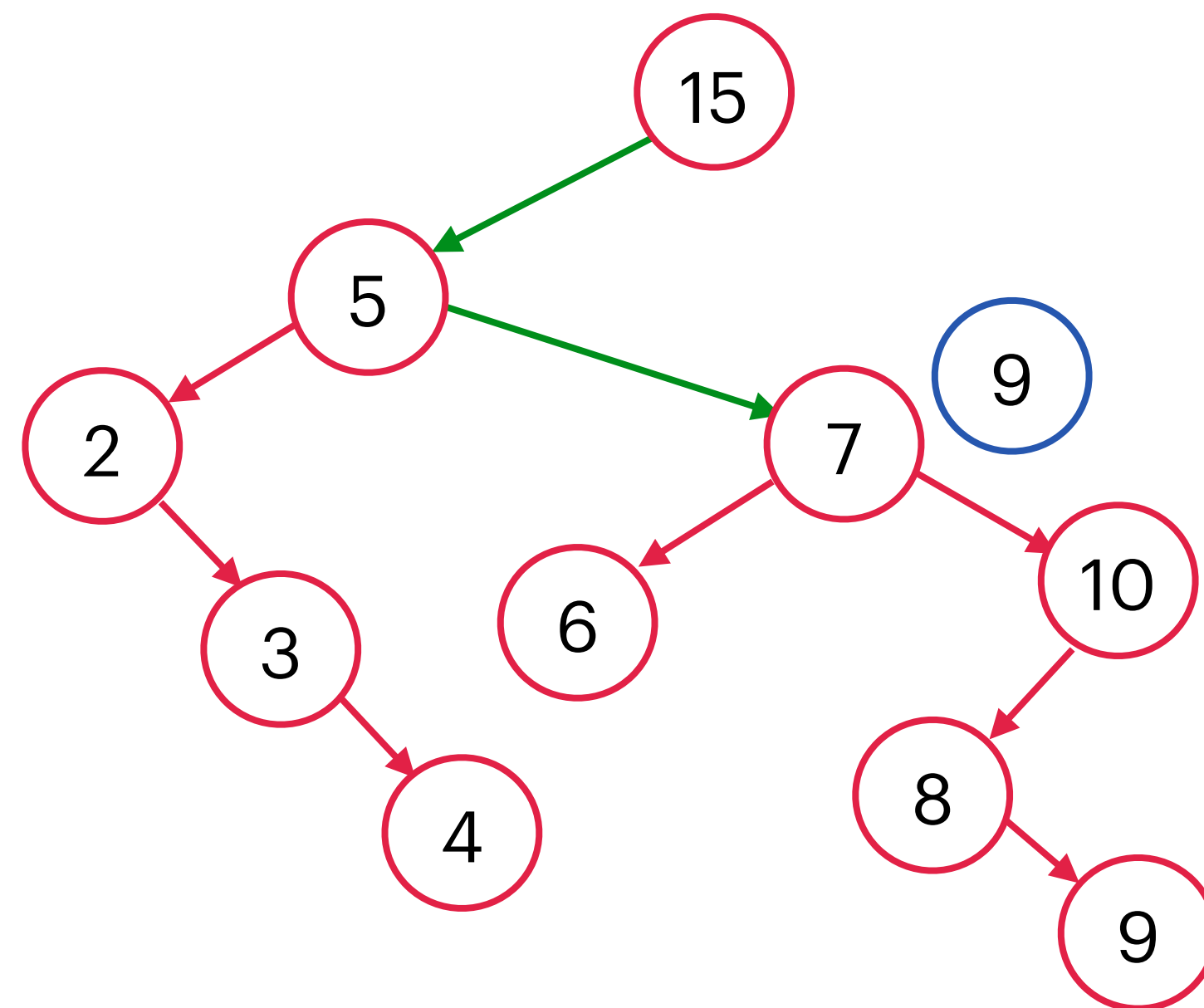
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

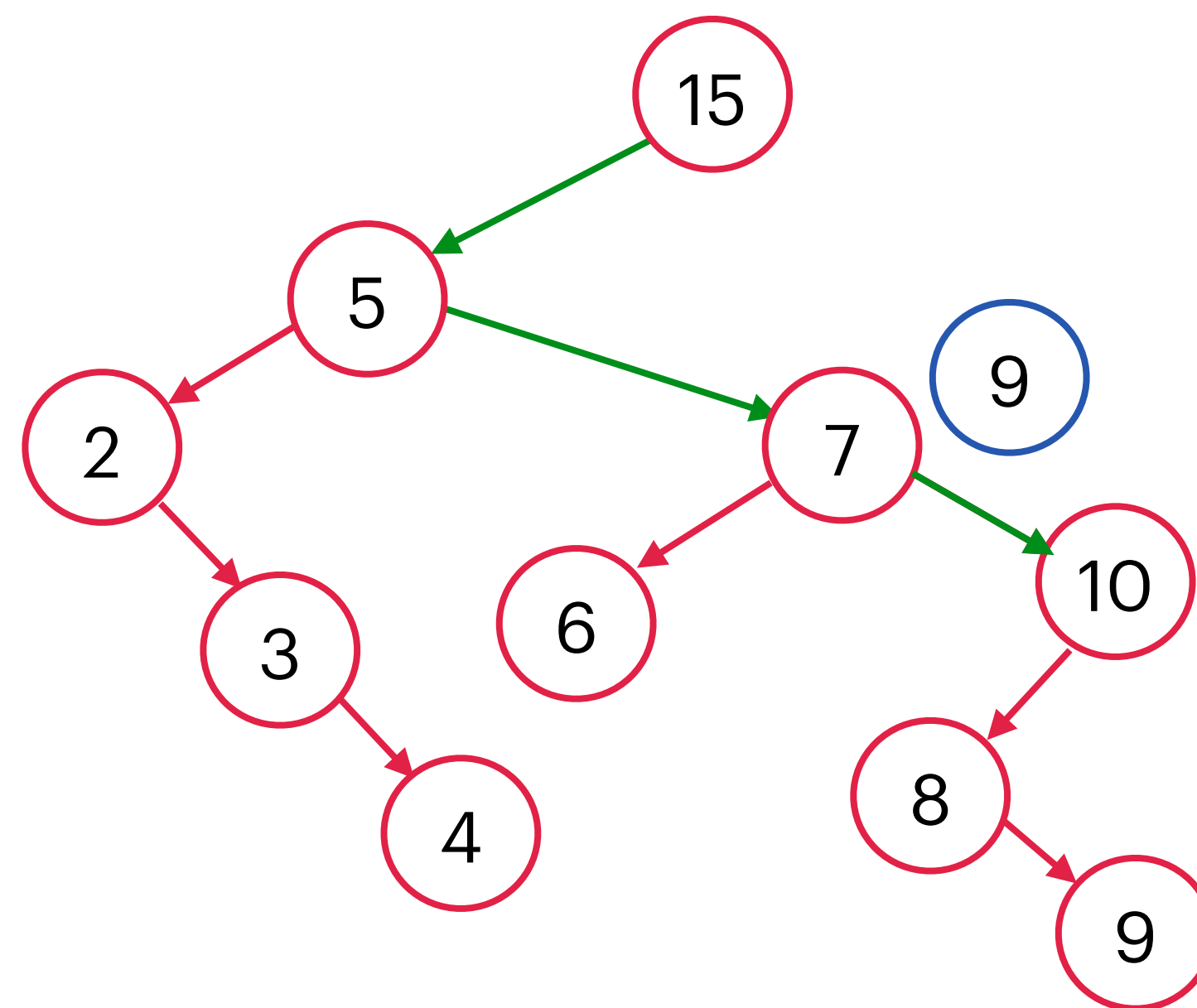
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonăți

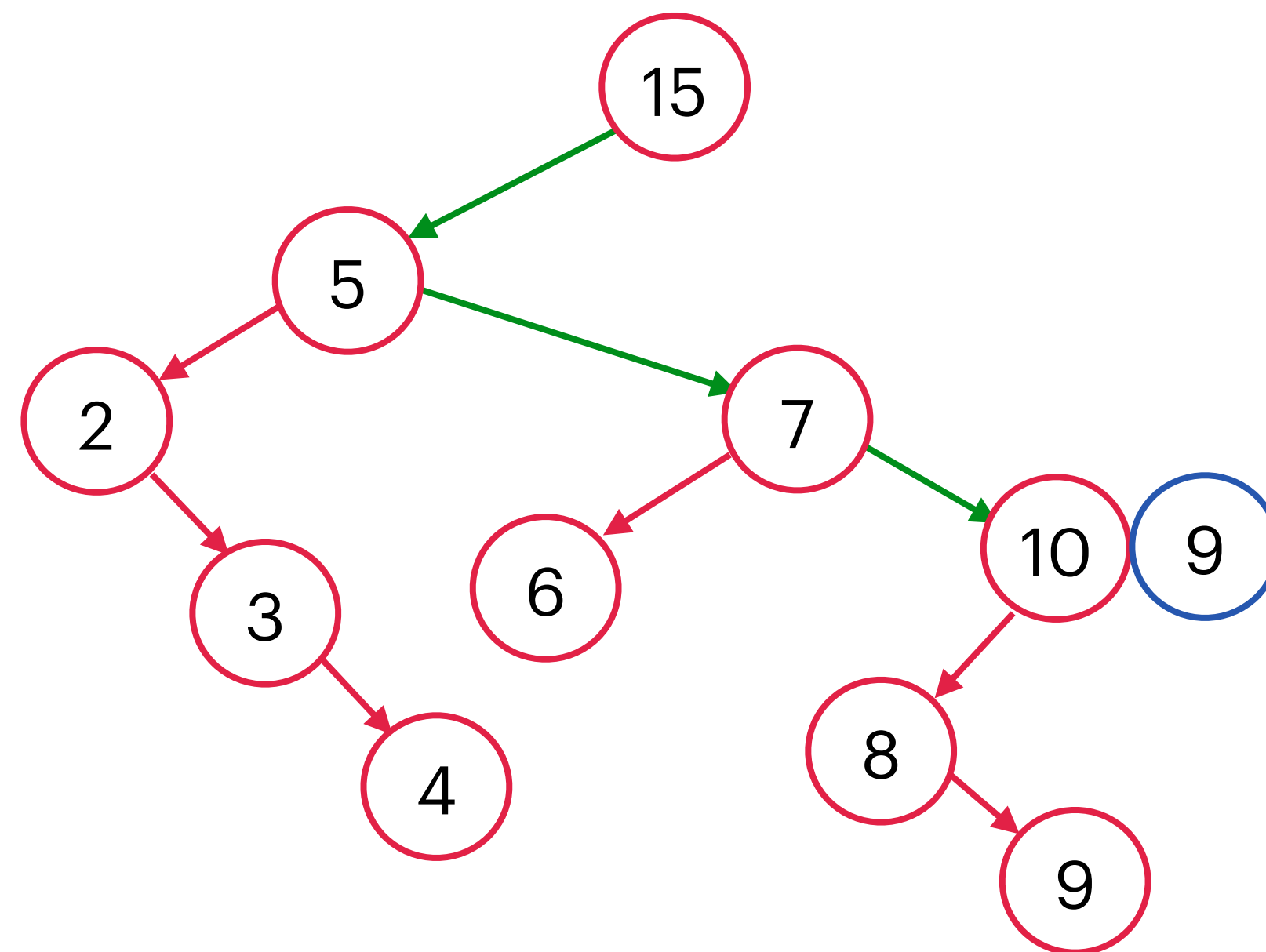
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

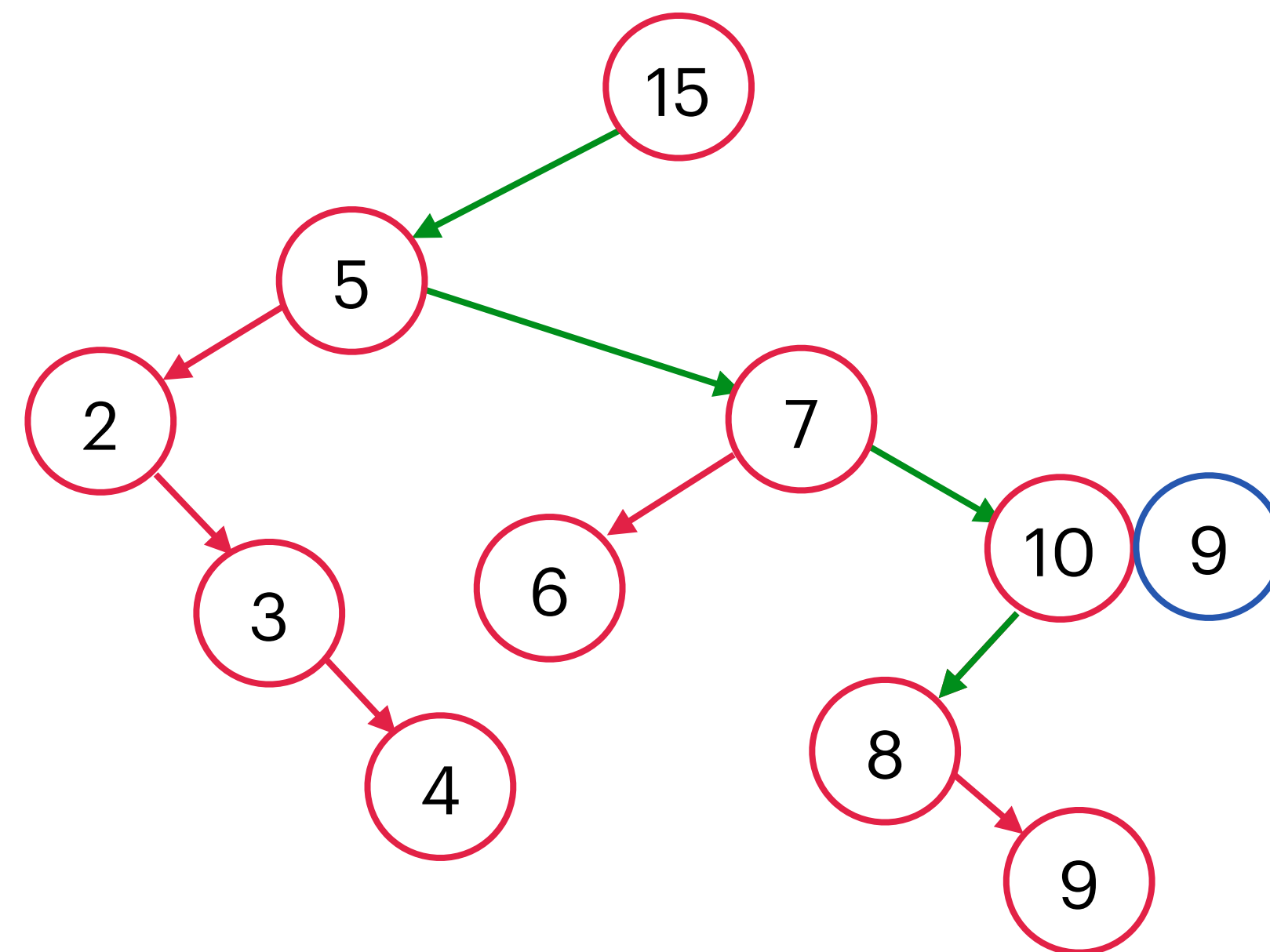
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

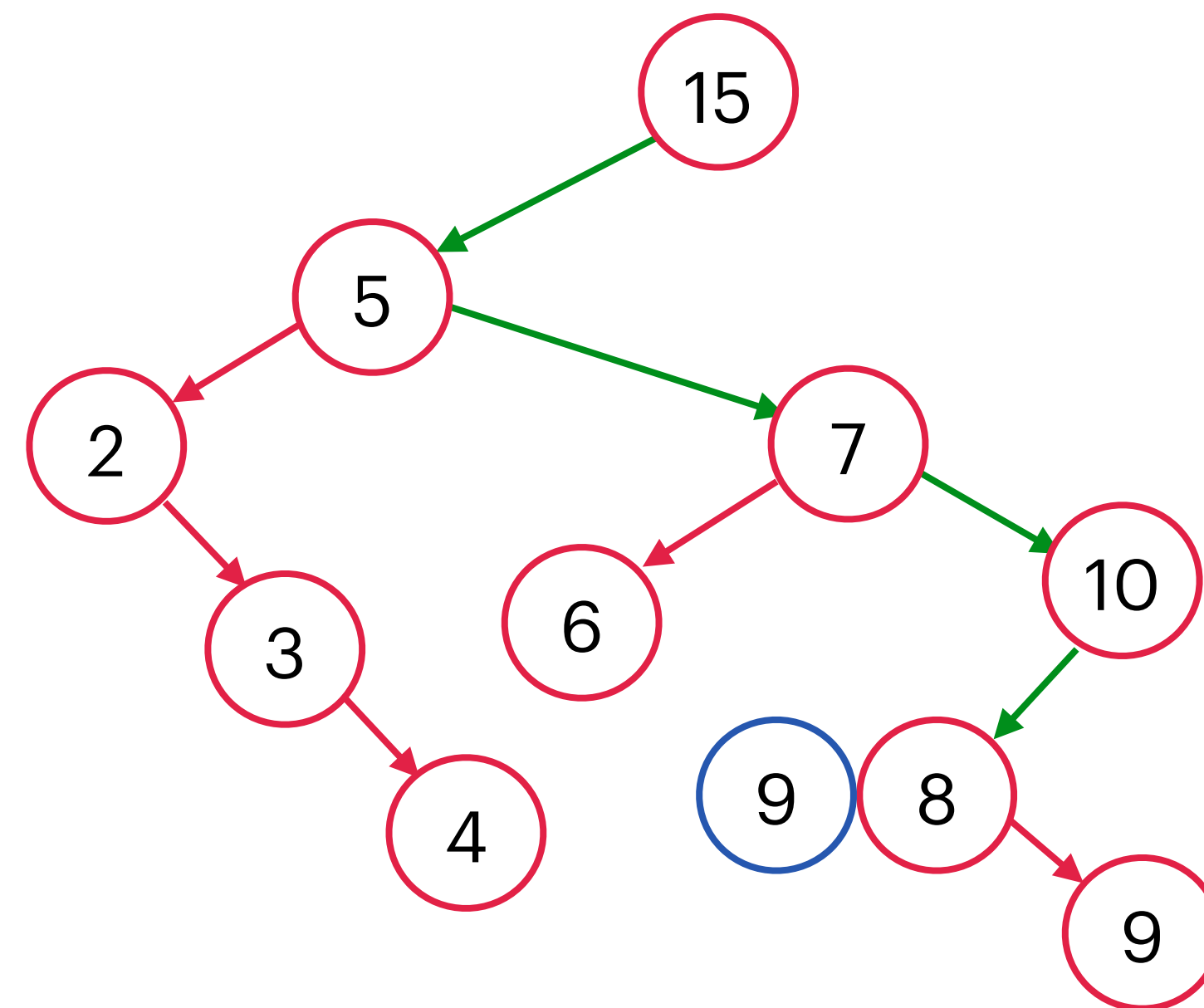
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

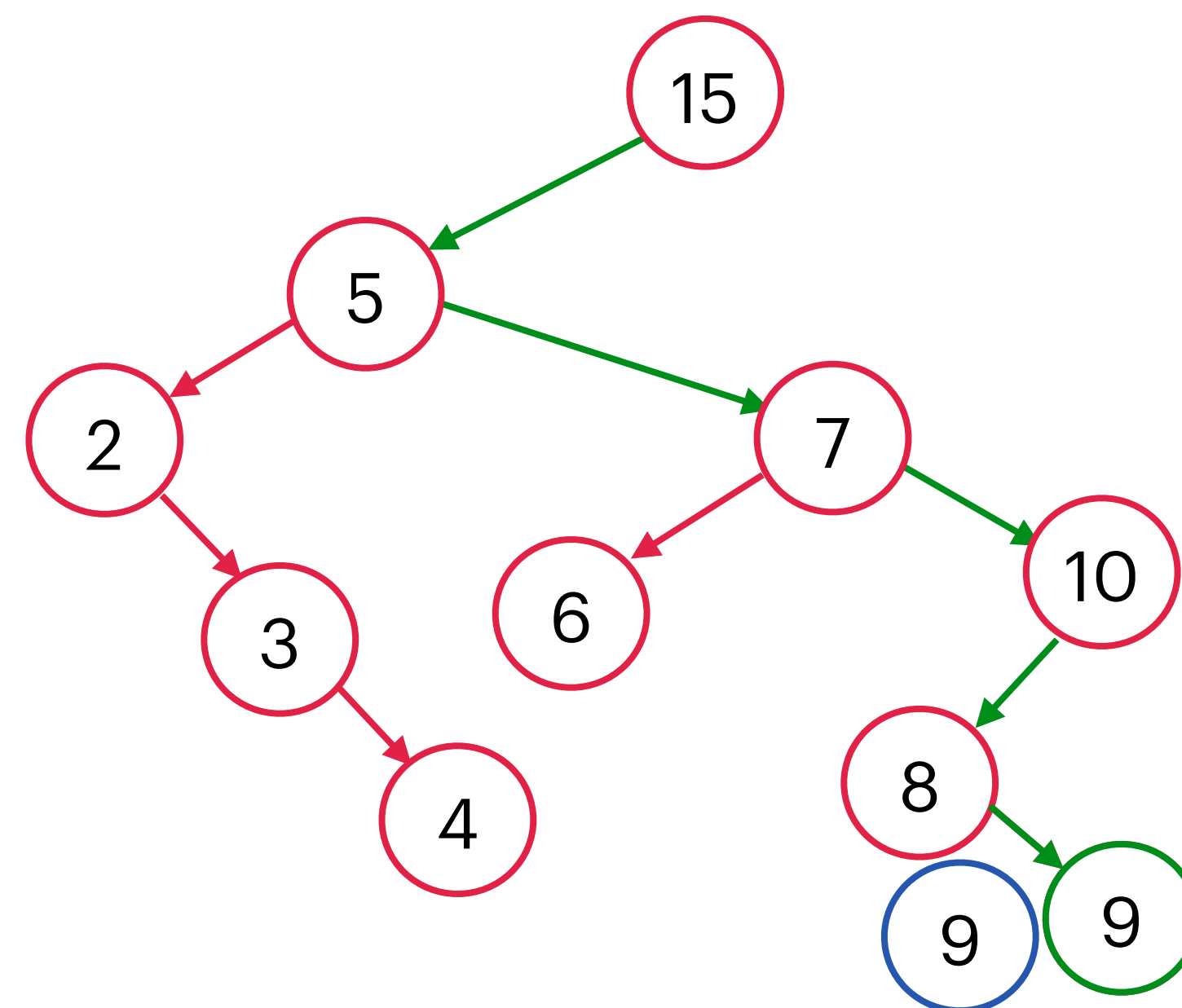
Prin arbore binar ordonat se înțelege un arbore binar care parcurs în inordine secvența cheilor este monoton crescătoare

Pentru orice nod n cu cheia c al unui arbore binar ordonat:

Toate nodurile din subarborele stâng al lui n au cheile mai mici sau egale cu c

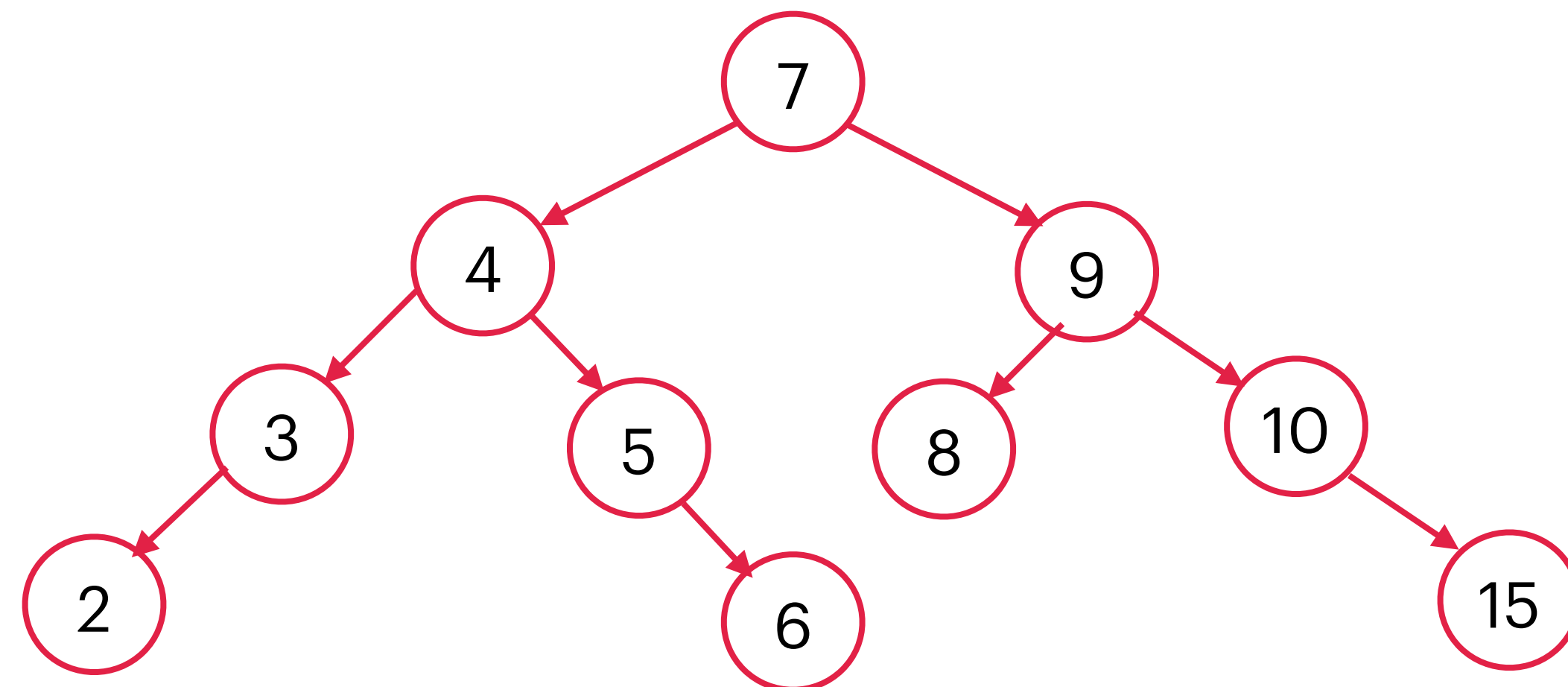
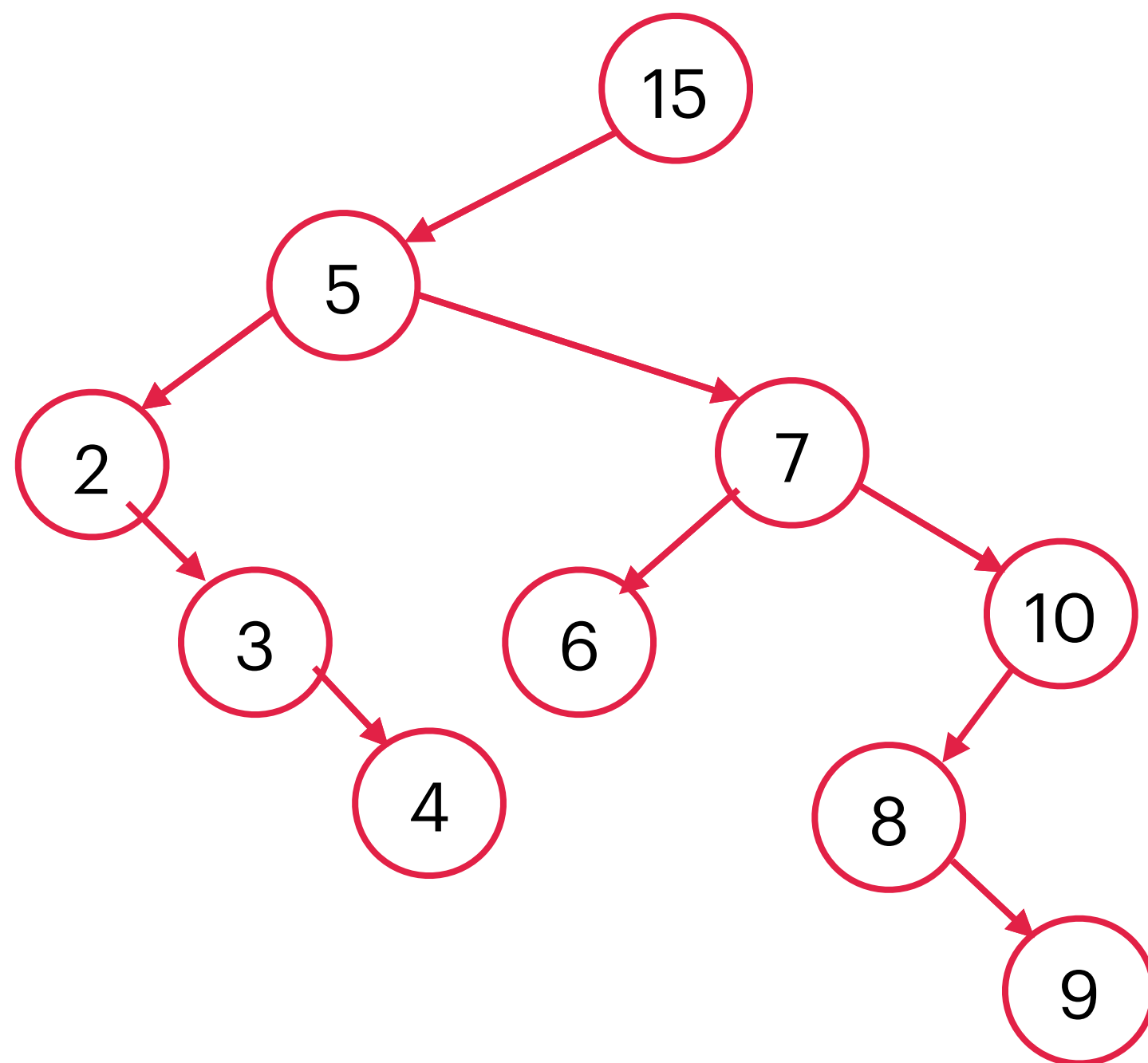
Toate nodurile din subarborele drept al lui n au cheile mai mari sau egale cu c

Căutarea în ABO se face începând cu rădăcina și se trece la fiul stâng sau fiul drept în funcție dacă cheia căutată este mai mică sau mai mare decât a nodului curent



Arbori Binari Ordonati

În general, înălțimea unui arbore binar ordonat nu este dată de numărul nodurilor sale



Arbori Binari Ordonati

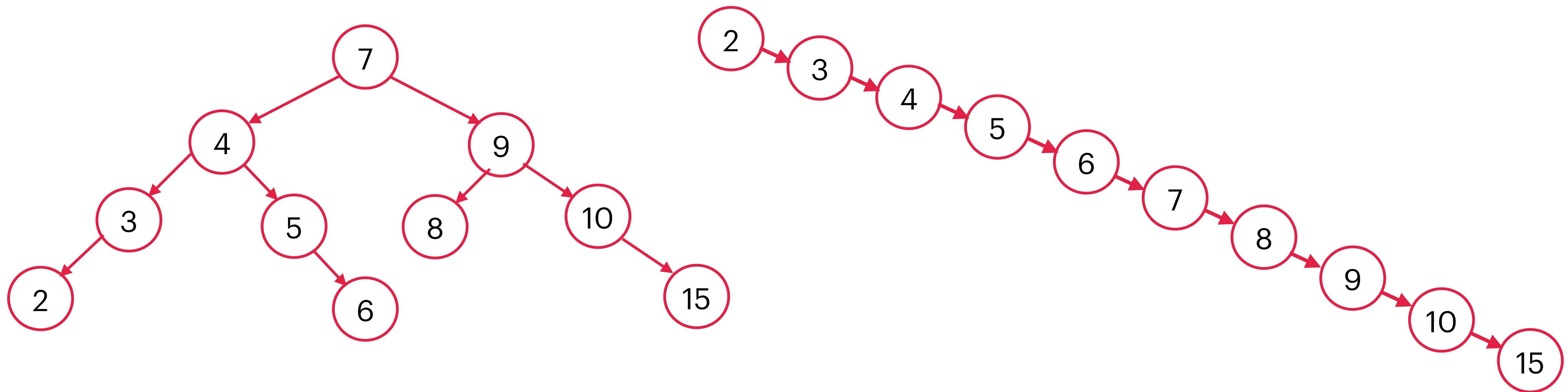
Un ABO are înălțimea minimă dacă fiecare nivel al său conține numărul maxim de noduri cu excepția posibilă a ultimului nivel

Numărul maxim de noduri al nivelului i este 2^{i-1} , rezulta că înălțimea minimă a unui arbore binar cu n noduri este $h_{\min} = \lceil \log_2 (n+1) \rceil$

Căutarea într-un ABO necesită aprox. $\log_2 n$ comparații.

Adevărat doar dacă arborele binar ordonat are înălțimea minimă

În cazul cel mai defavorabil căutarea are n comparații, când arborele este o listă simplu înlănțuită



Arbori Binari Ordonati

Insertia nodurilor în ABO

În urma inserției, arborele trebuie să rămână ordonat.

Valoare care se inserează se compară începând de la rădăcină

Dacă valoarea este mai mică decât valoarea nodului curent, se continuă cu fiul stâng

Dacă valoarea este mai mare decât valoarea nodului curent, se continuă cu fiul drept

Când se găsește un loc disponibil, se crează un nod cu valoarea de inserat în poziția respectivă

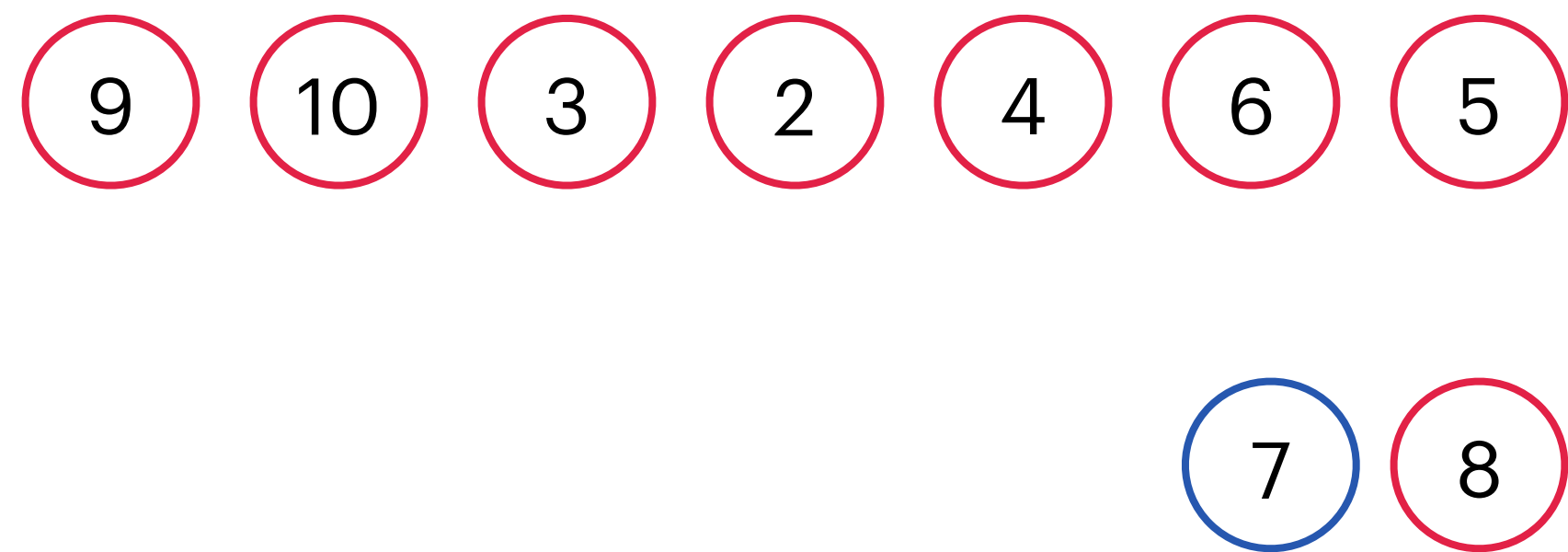
Arbori Binari Ordonati



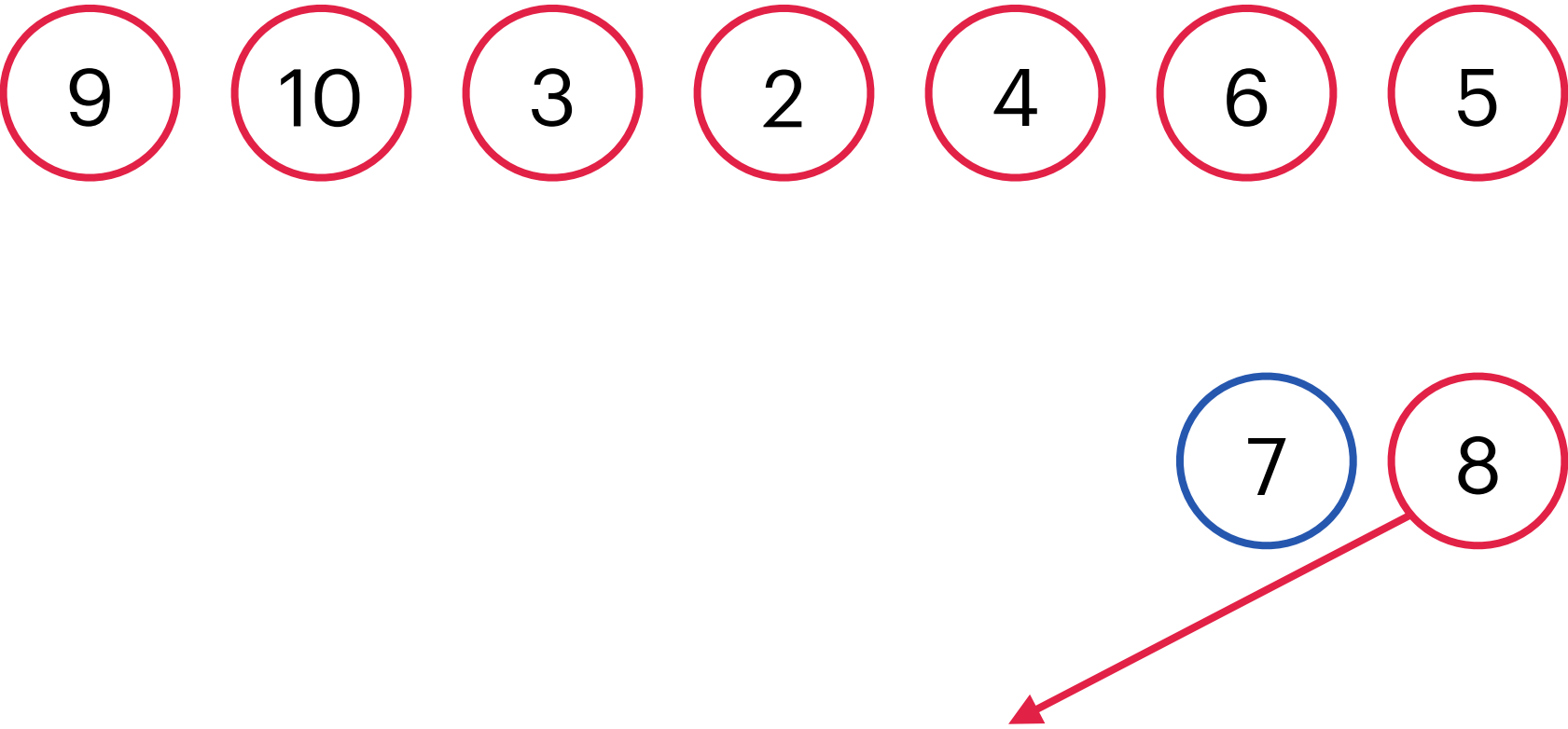
Arbori Binari Ordonati



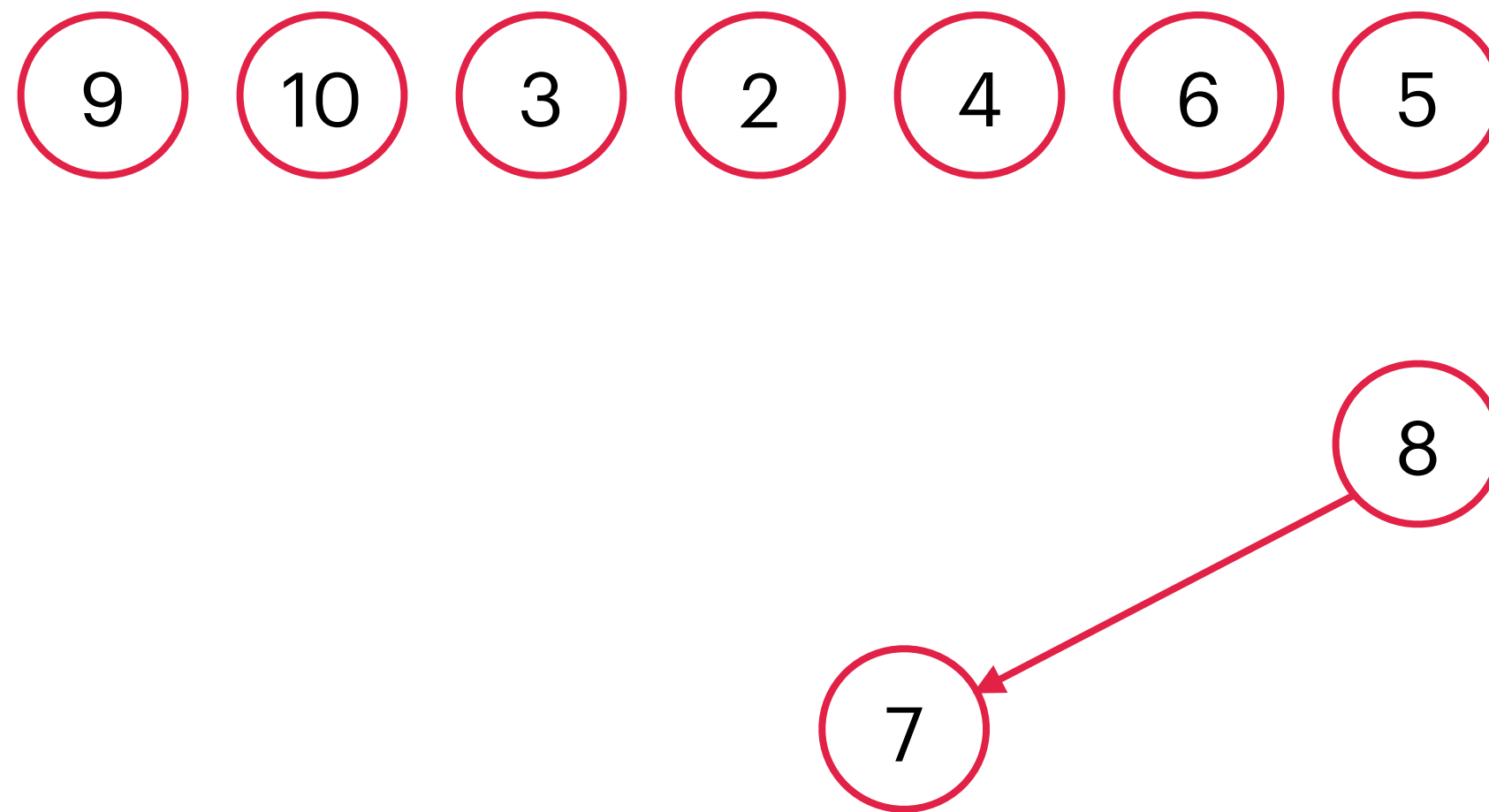
Arbori Binari Ordonati



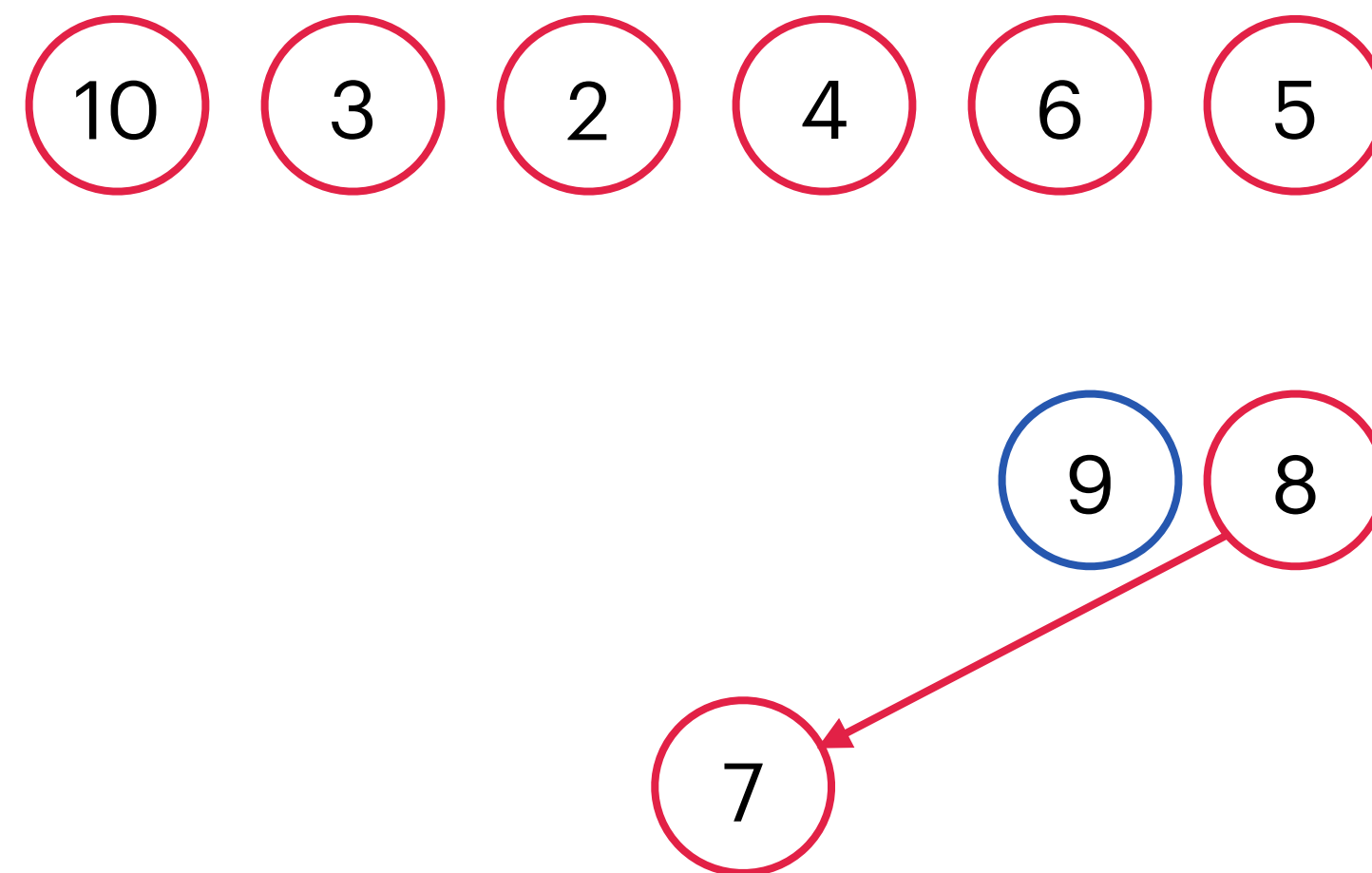
Arbori Binari Ordonati



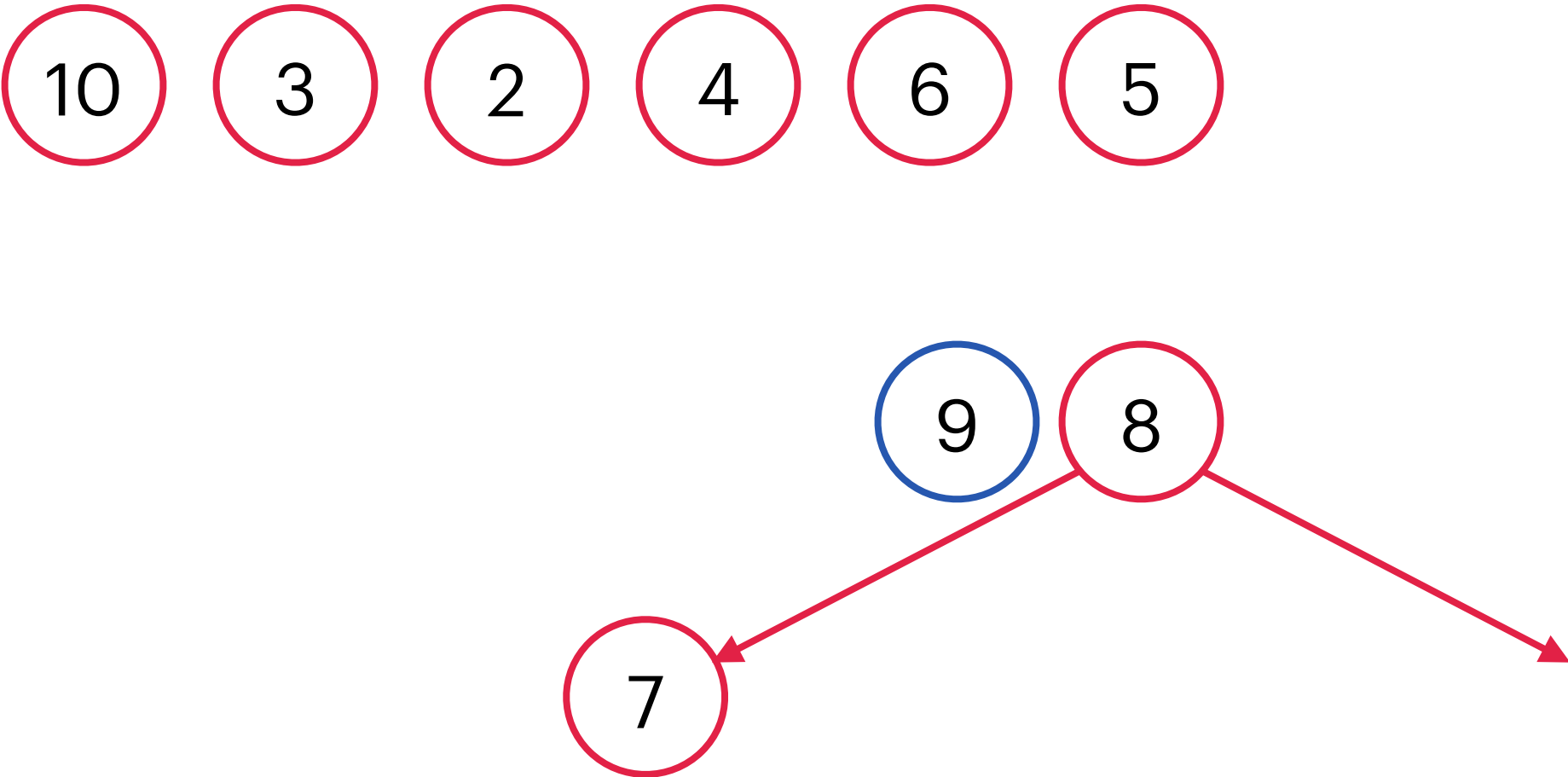
Arbori Binari Ordonati



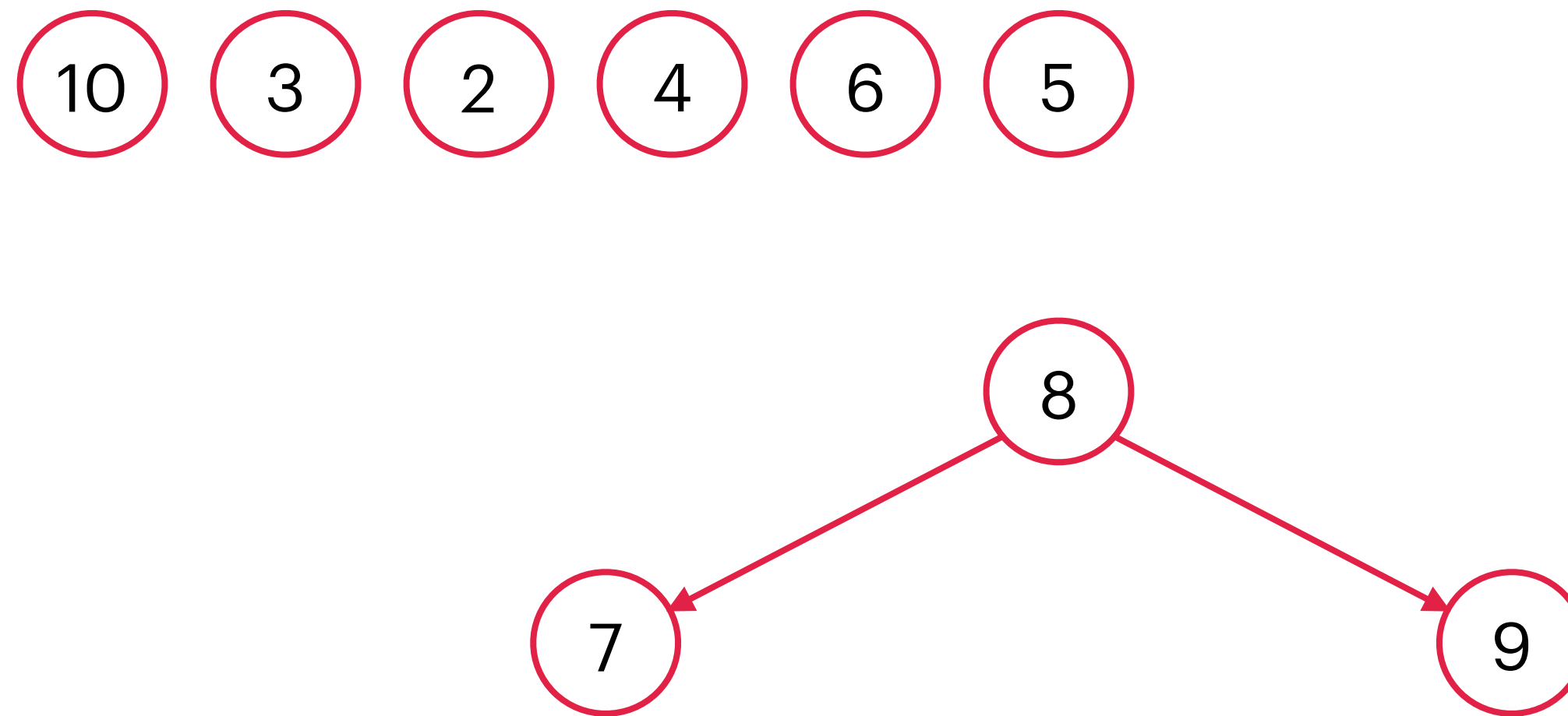
Arbori Binari Ordonati



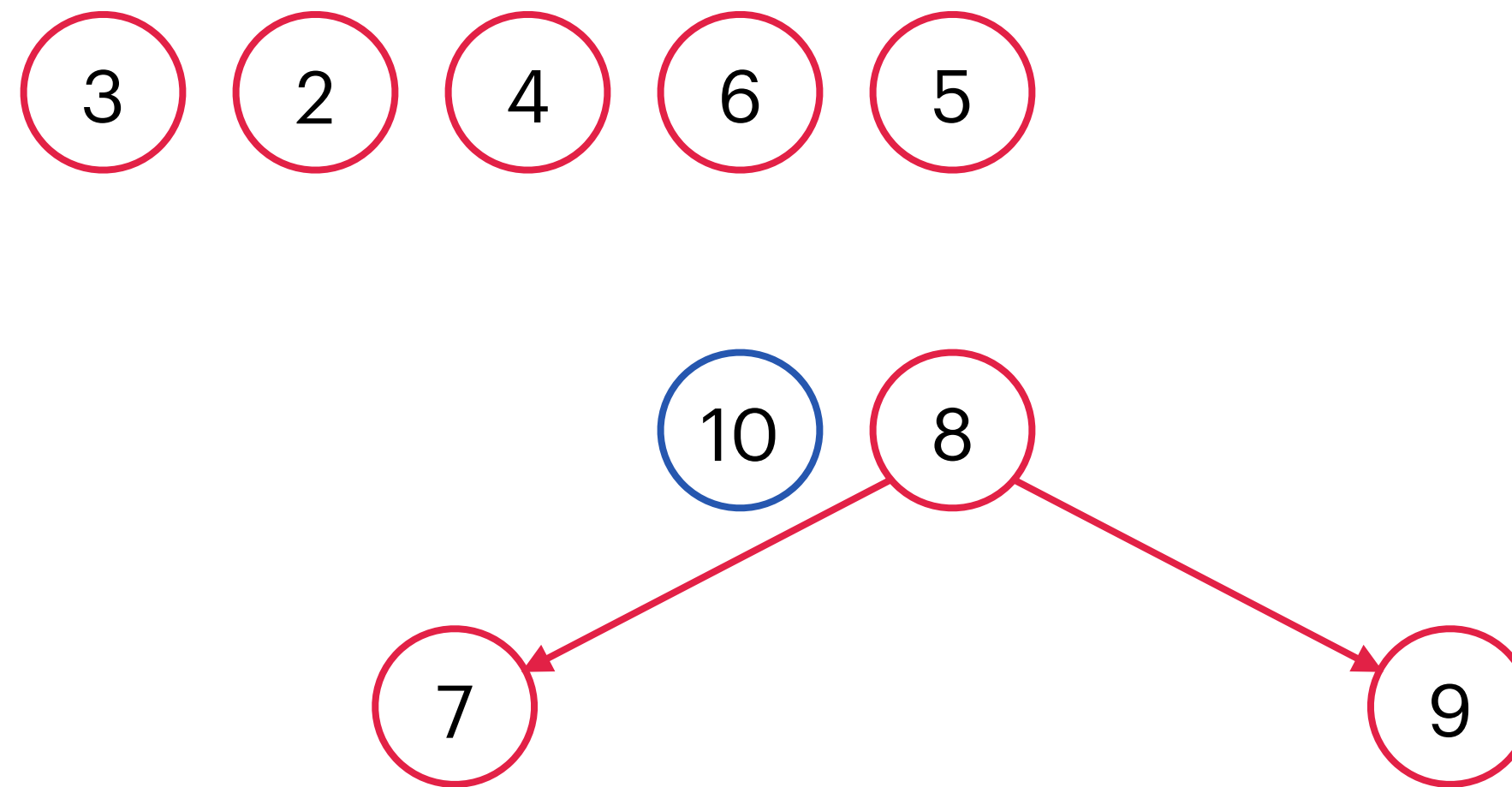
Arbori Binari Ordonati



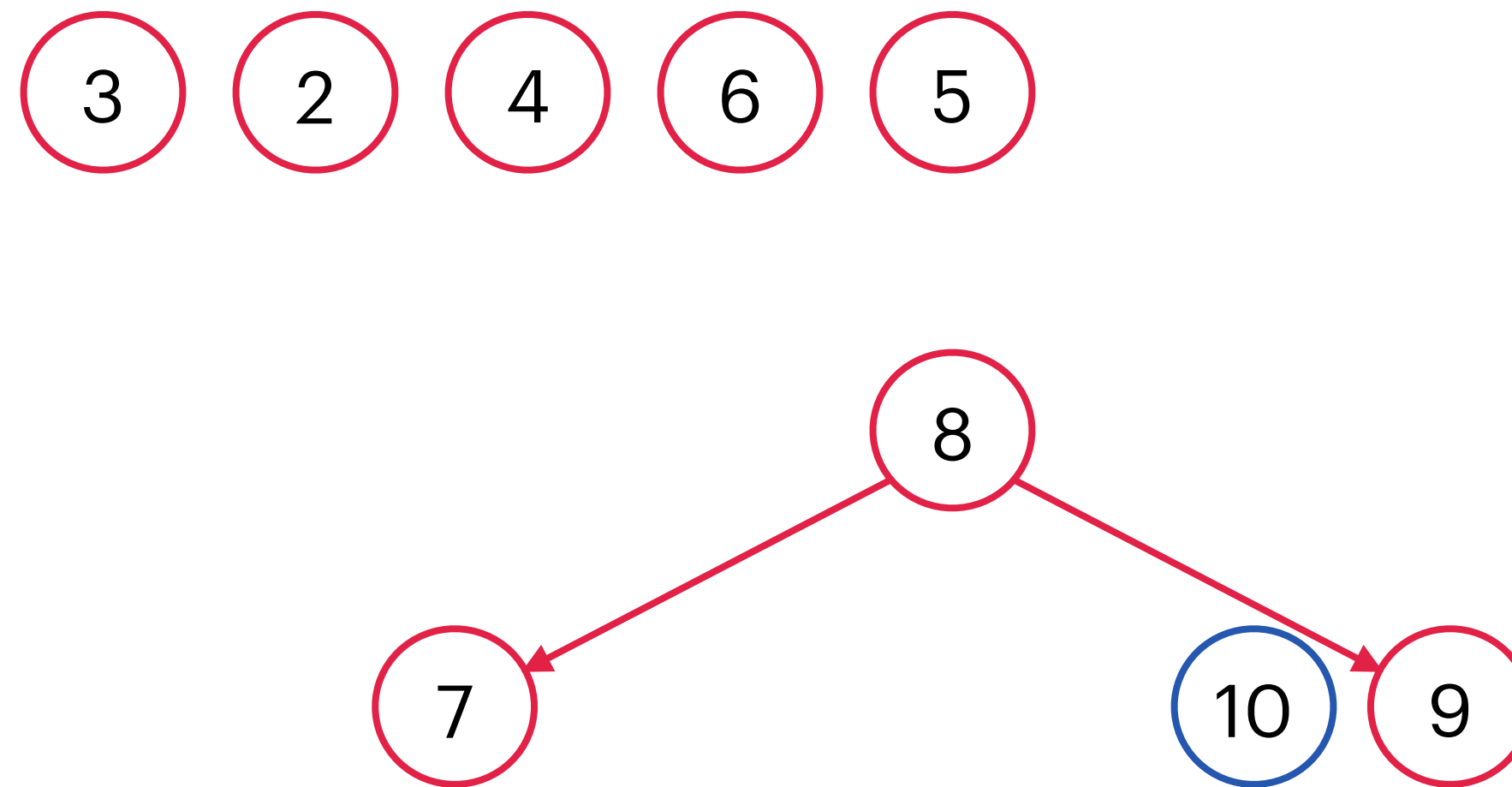
Arbori Binari Ordonati



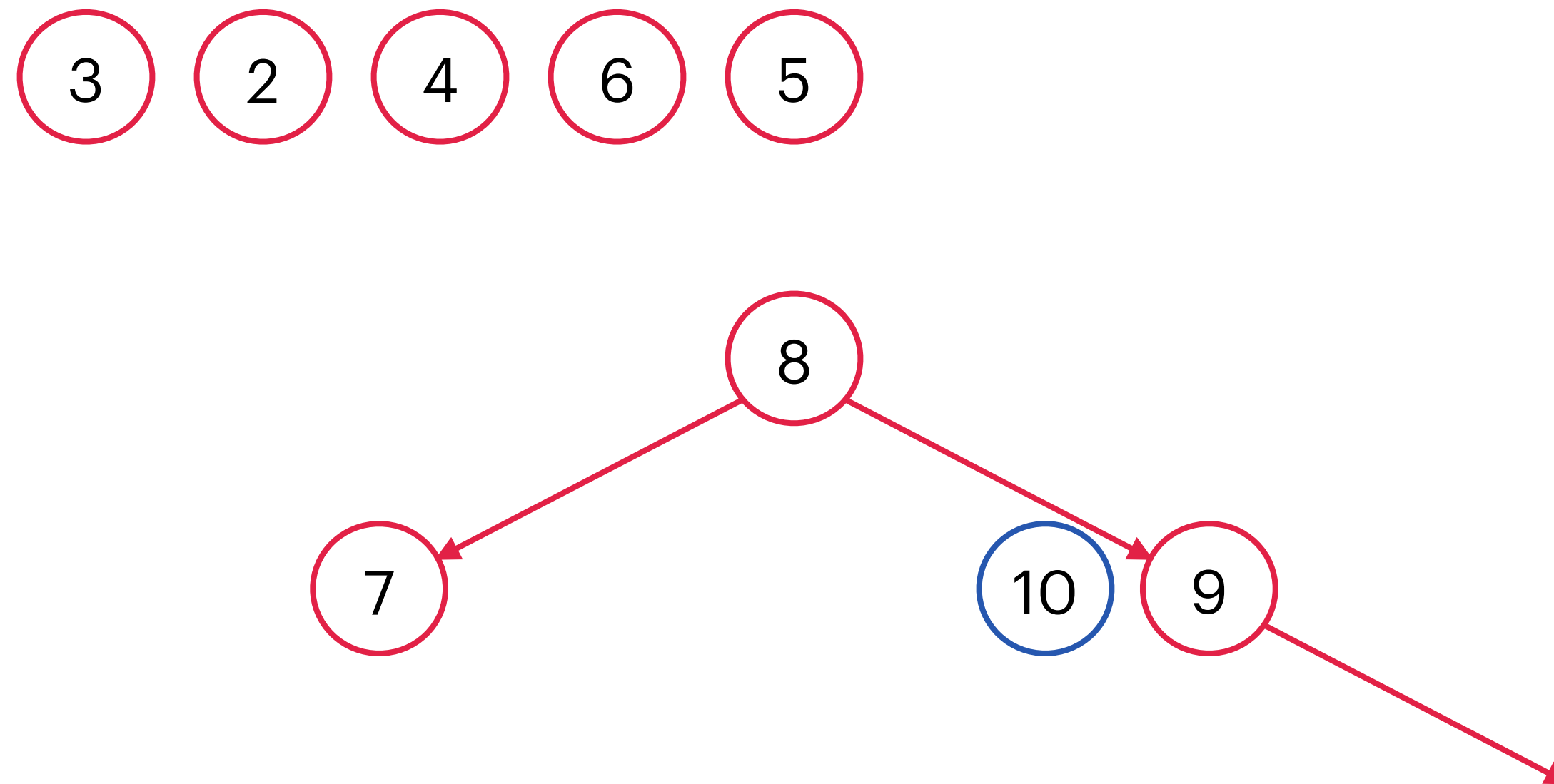
Arbori Binari Ordonati



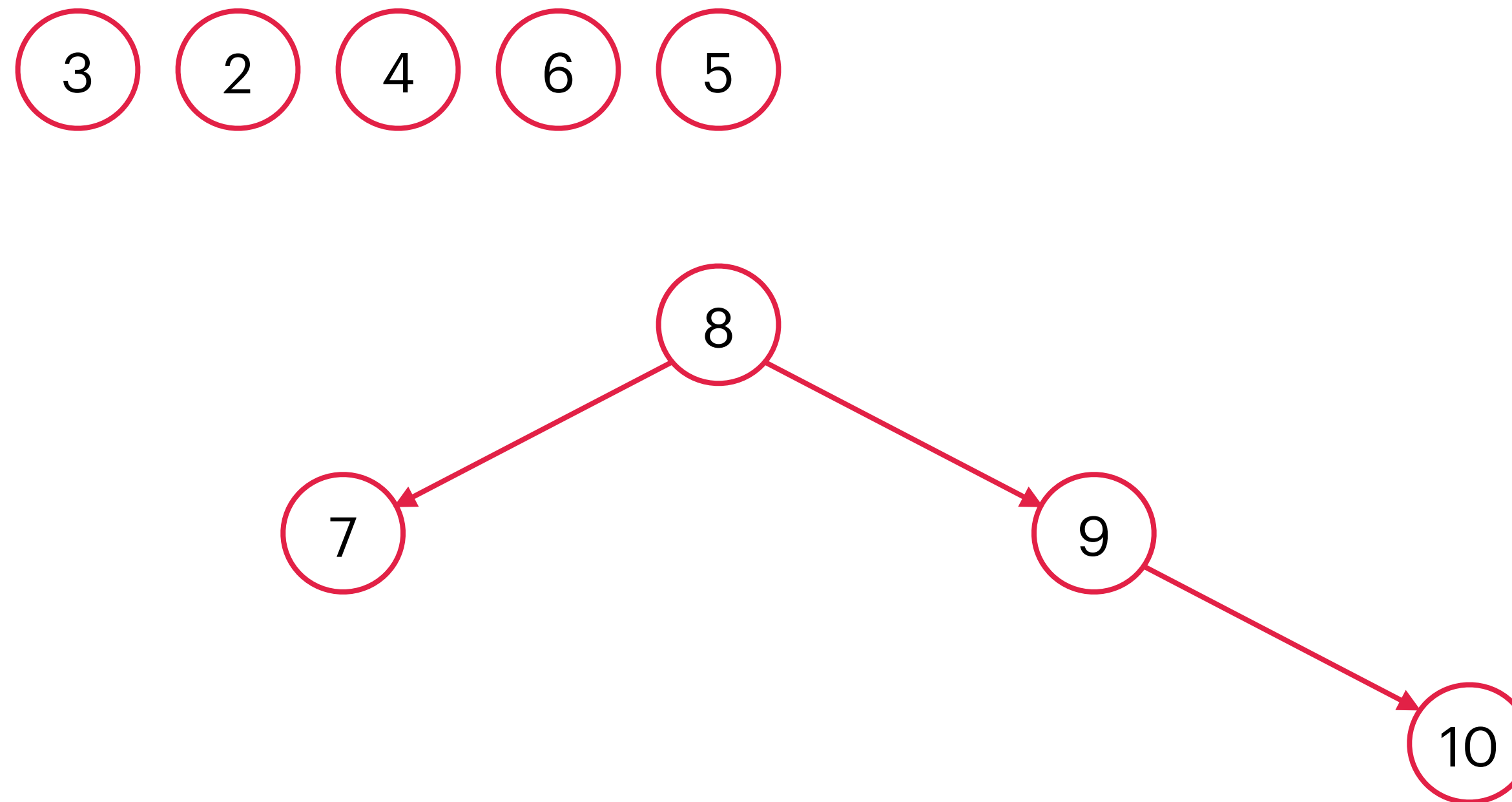
Arbori Binari Ordonati



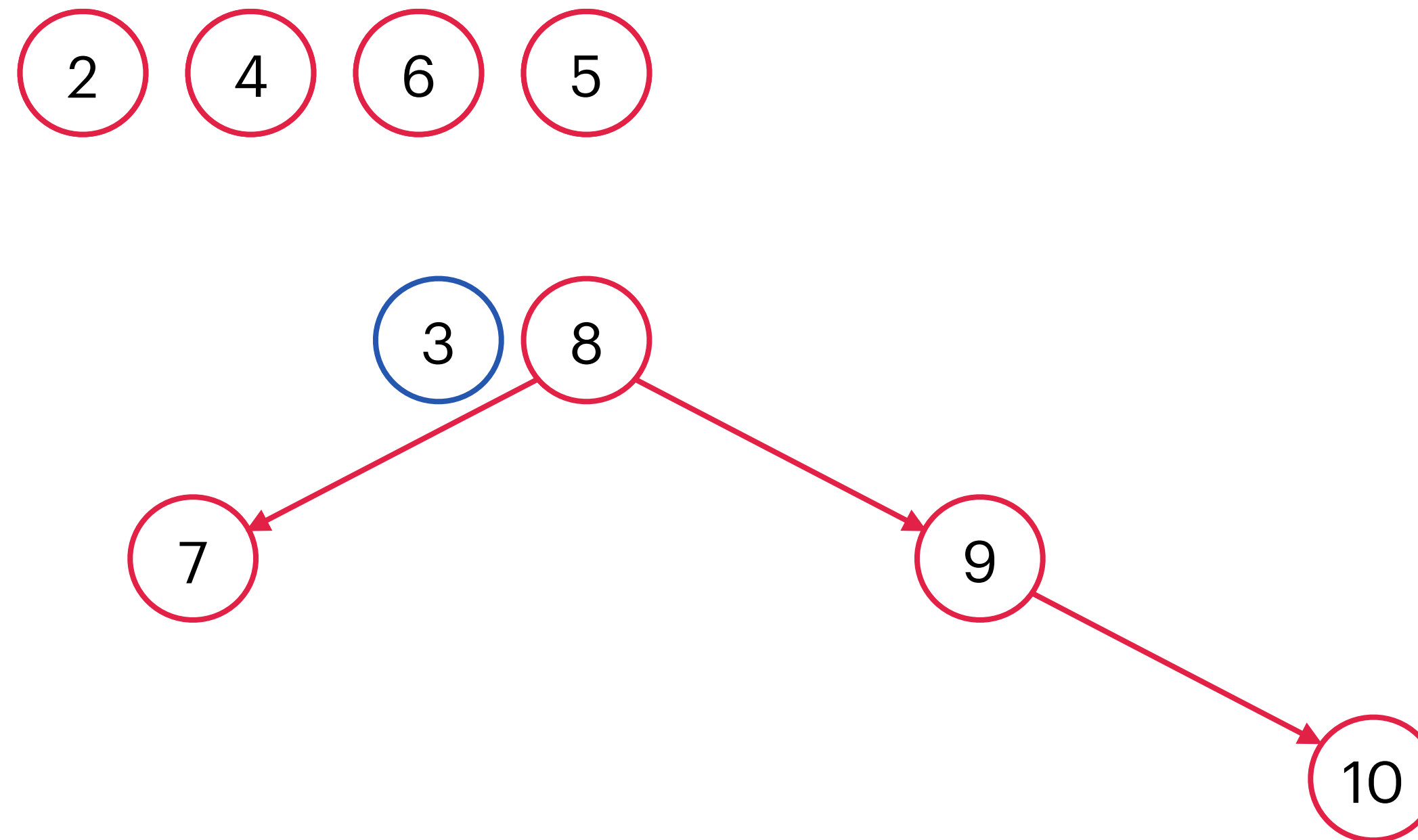
Arbori Binari Ordonati



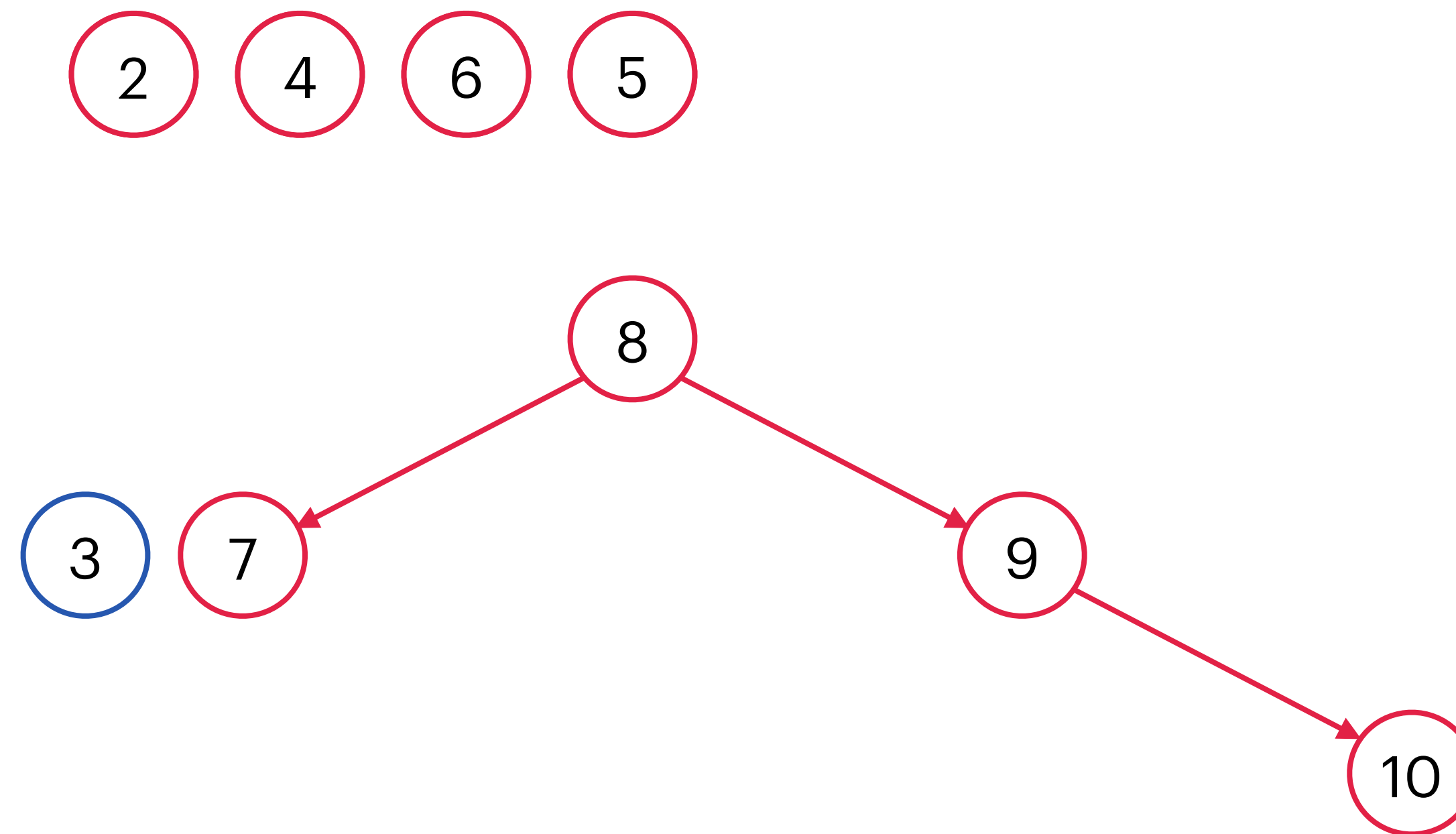
Arbori Binari Ordonati



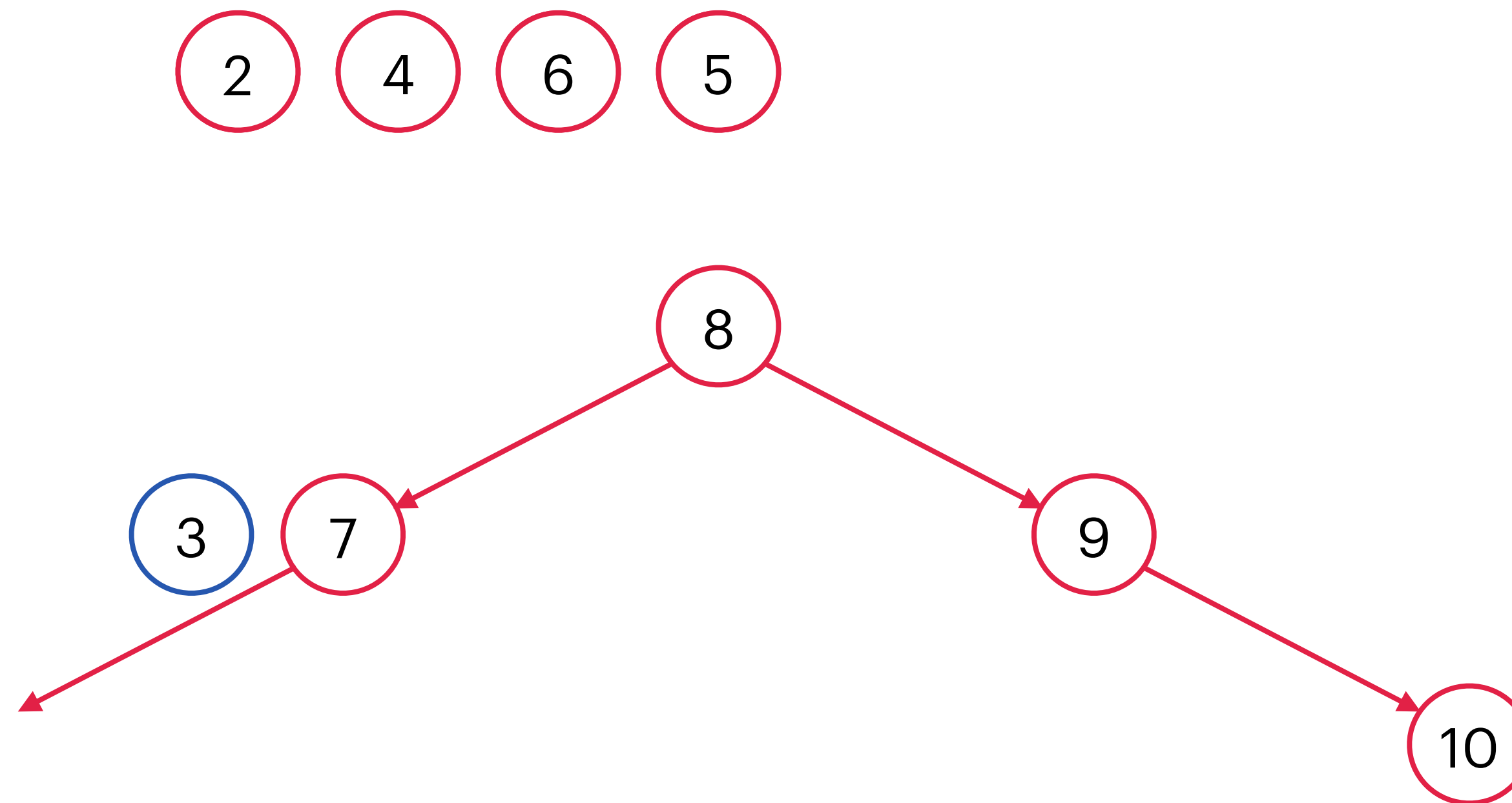
Arbori Binari Ordonati



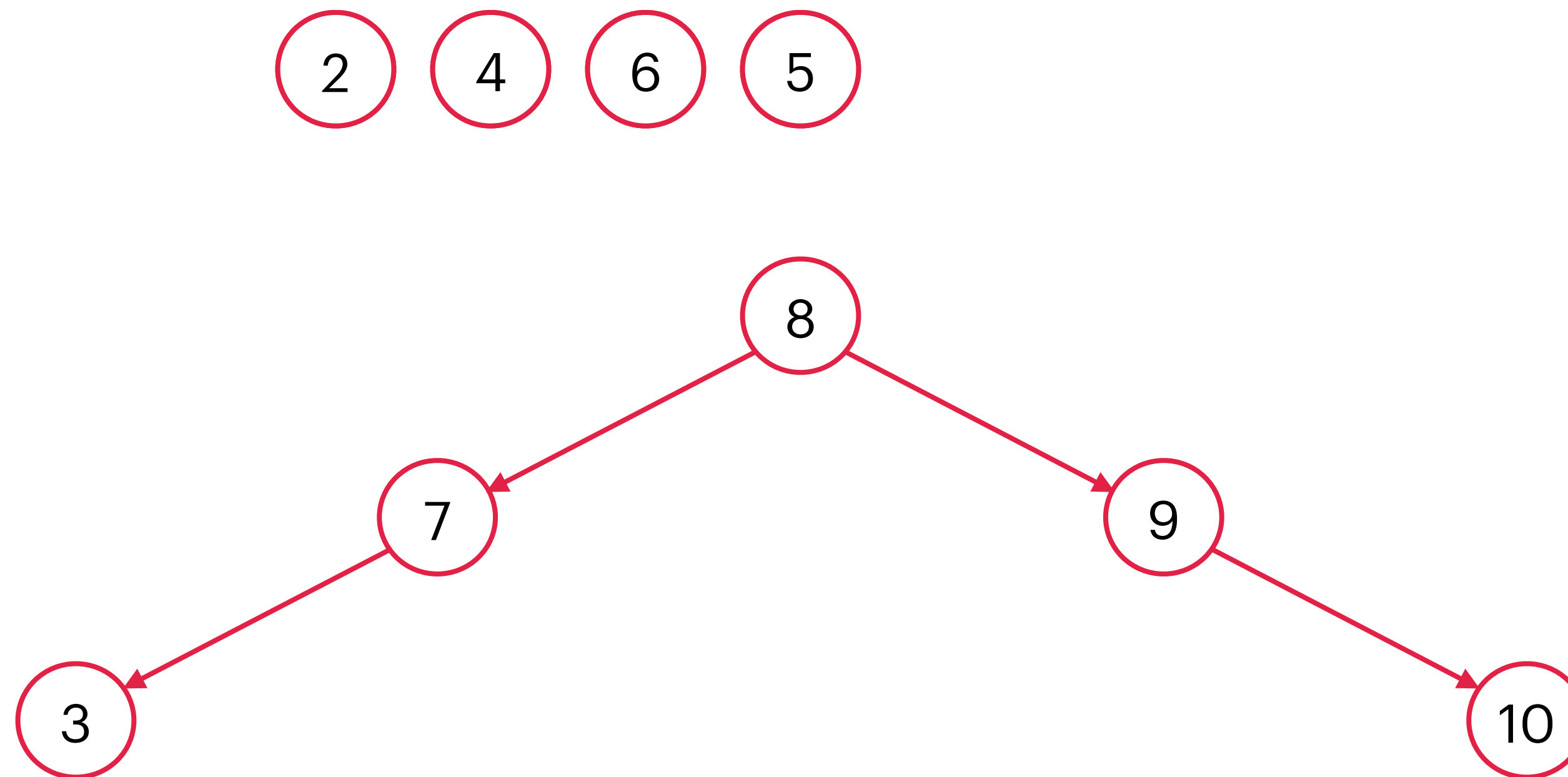
Arbori Binari Ordonati



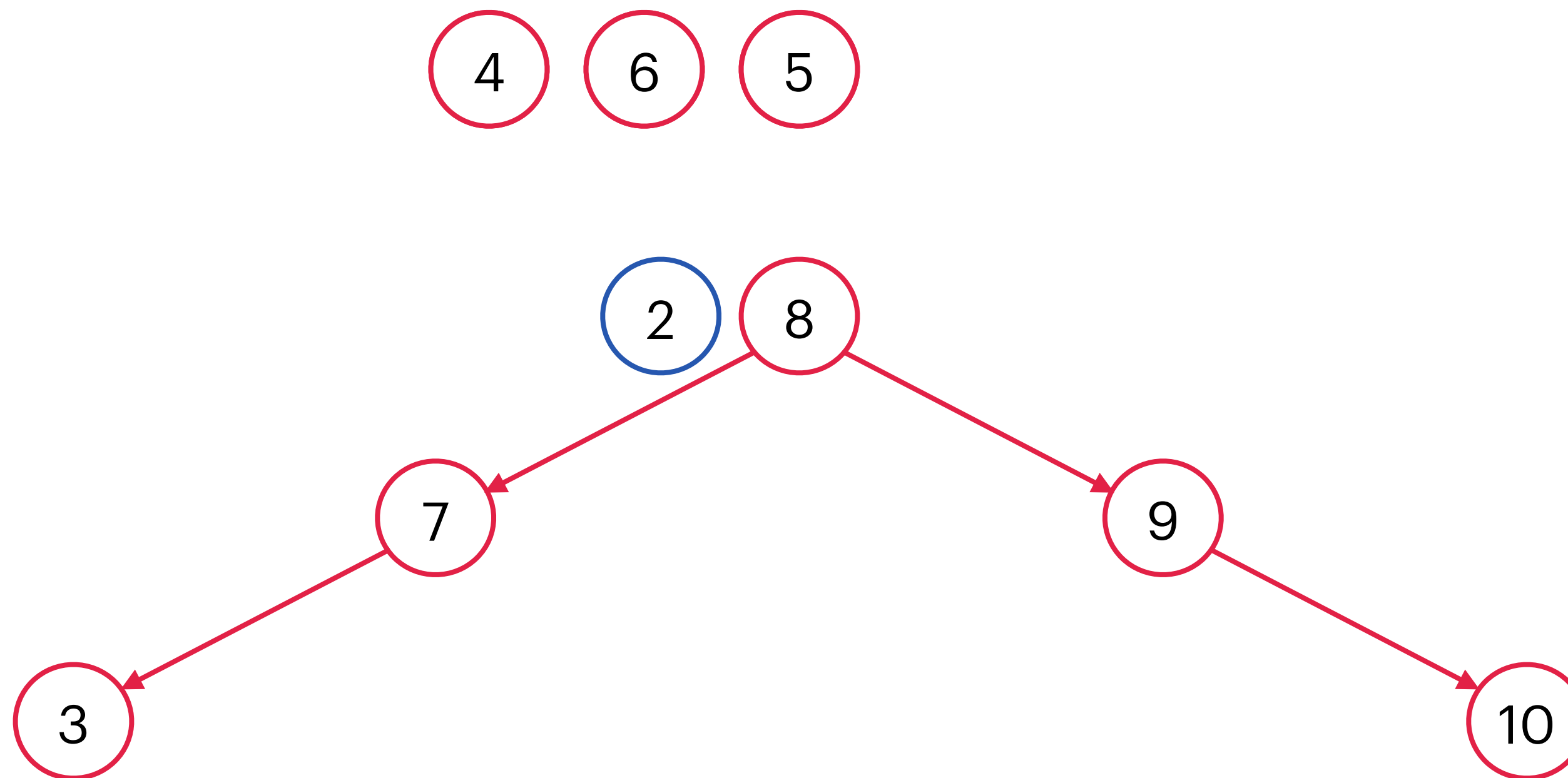
Arbori Binari Ordonati



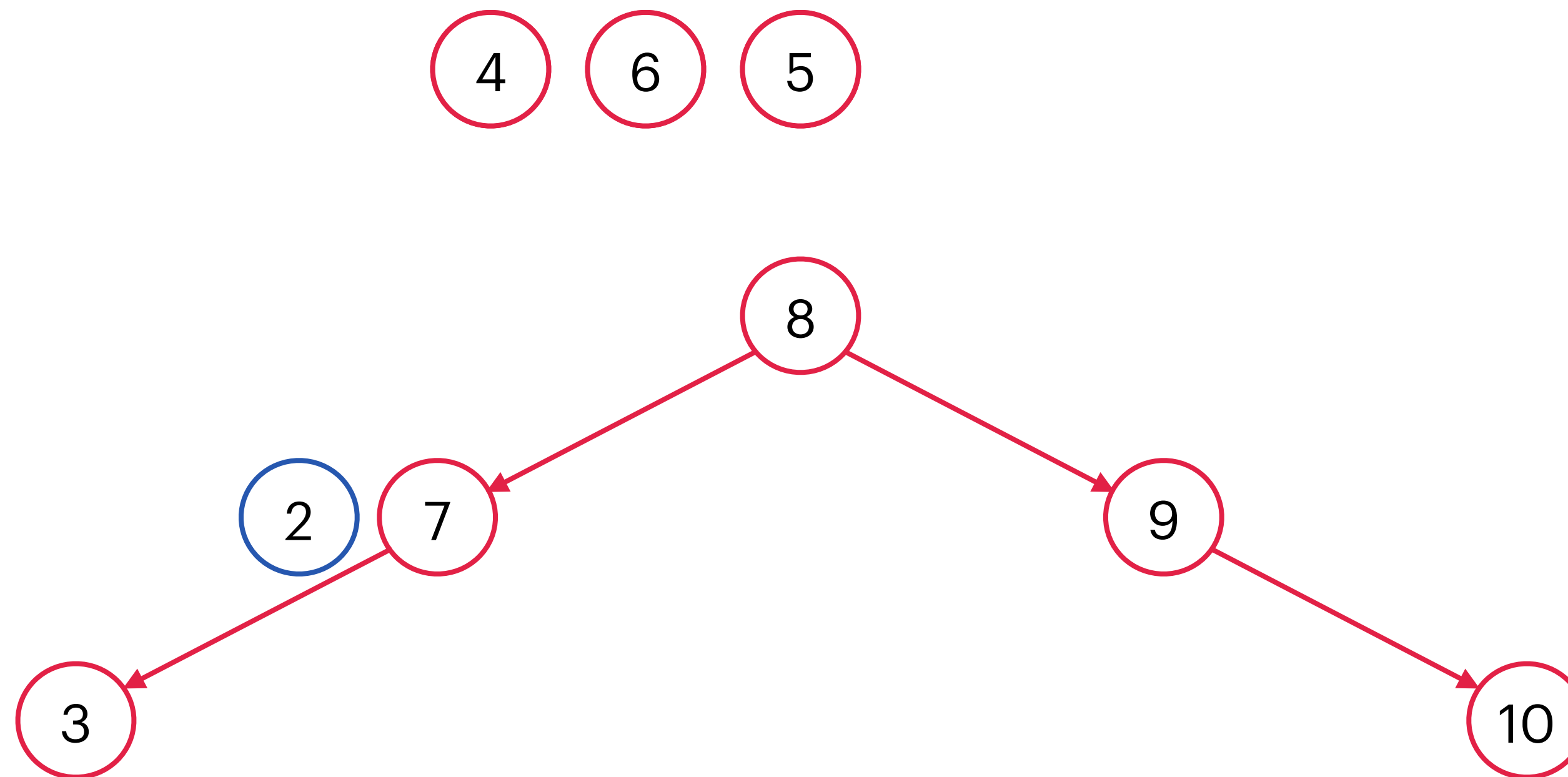
Arbori Binari Ordonati



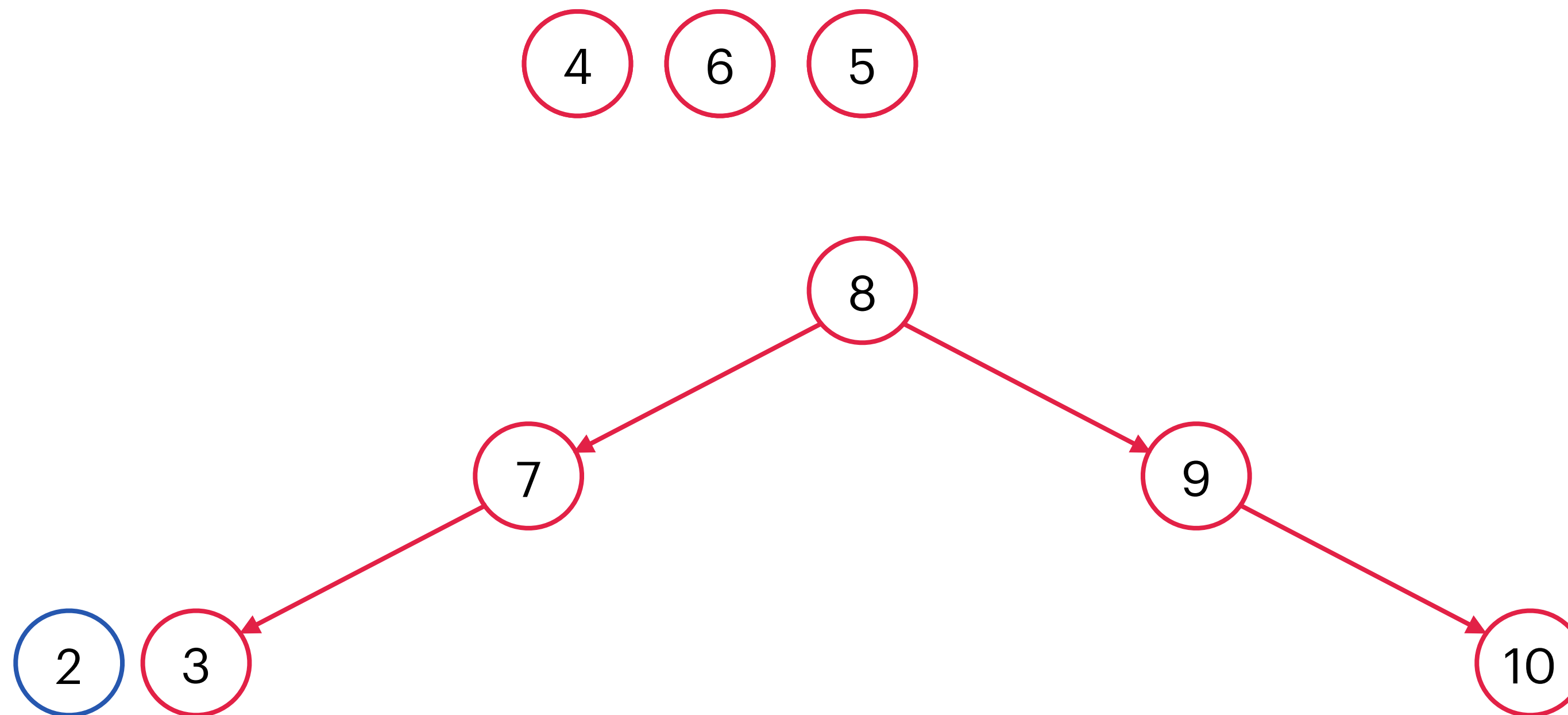
Arbori Binari Ordonati



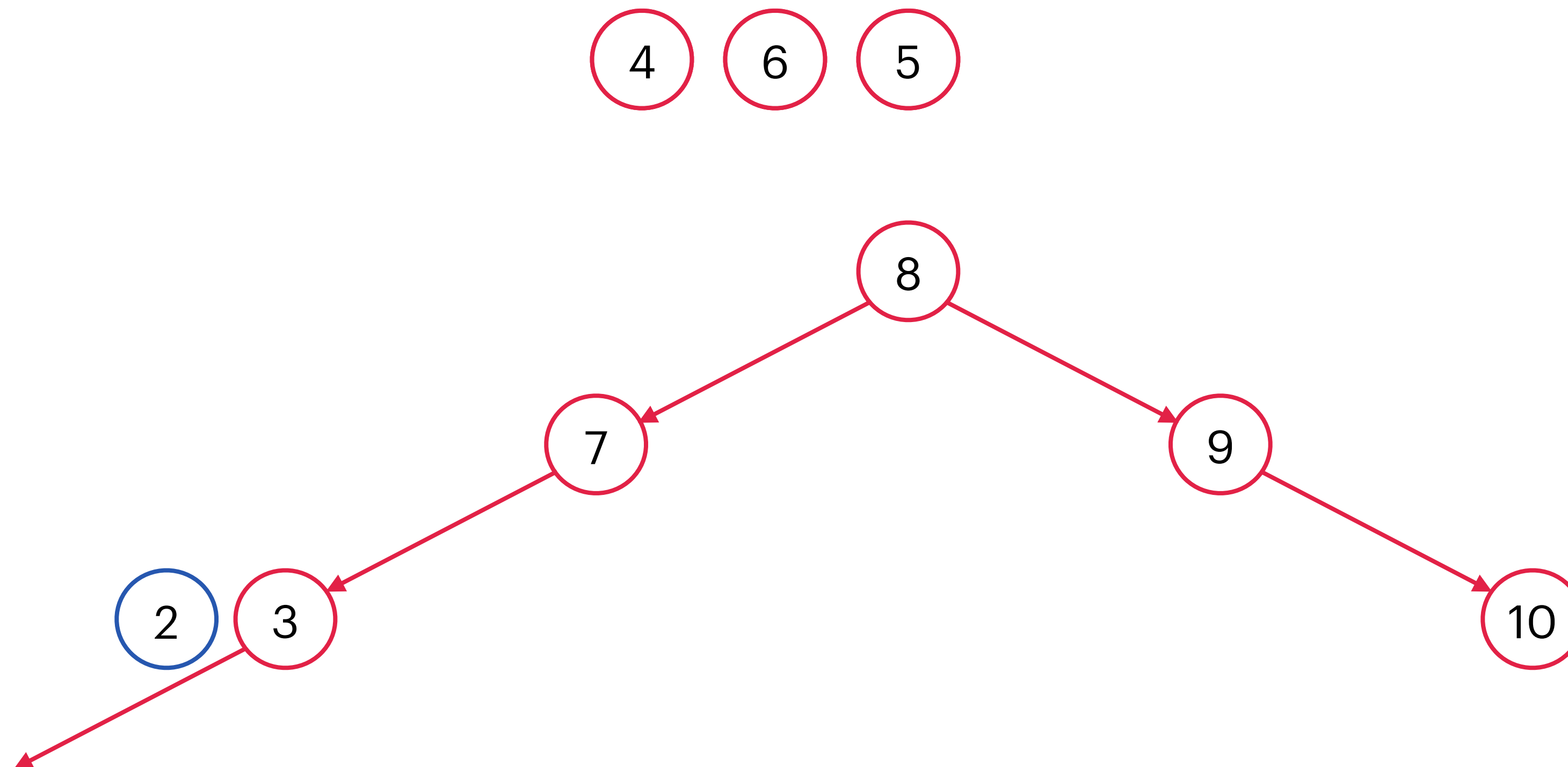
Arbori Binari Ordonati



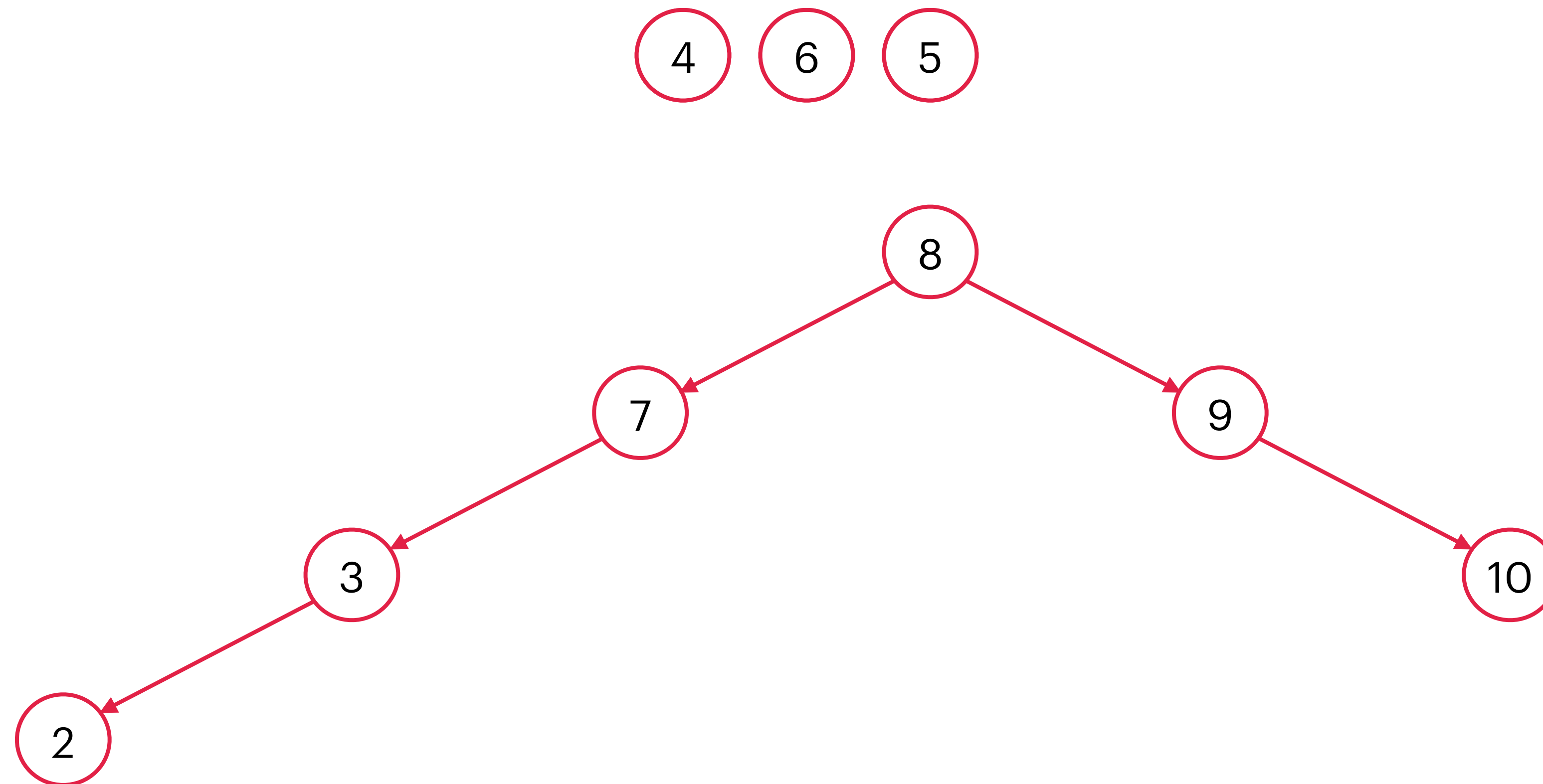
Arbori Binari Ordonati



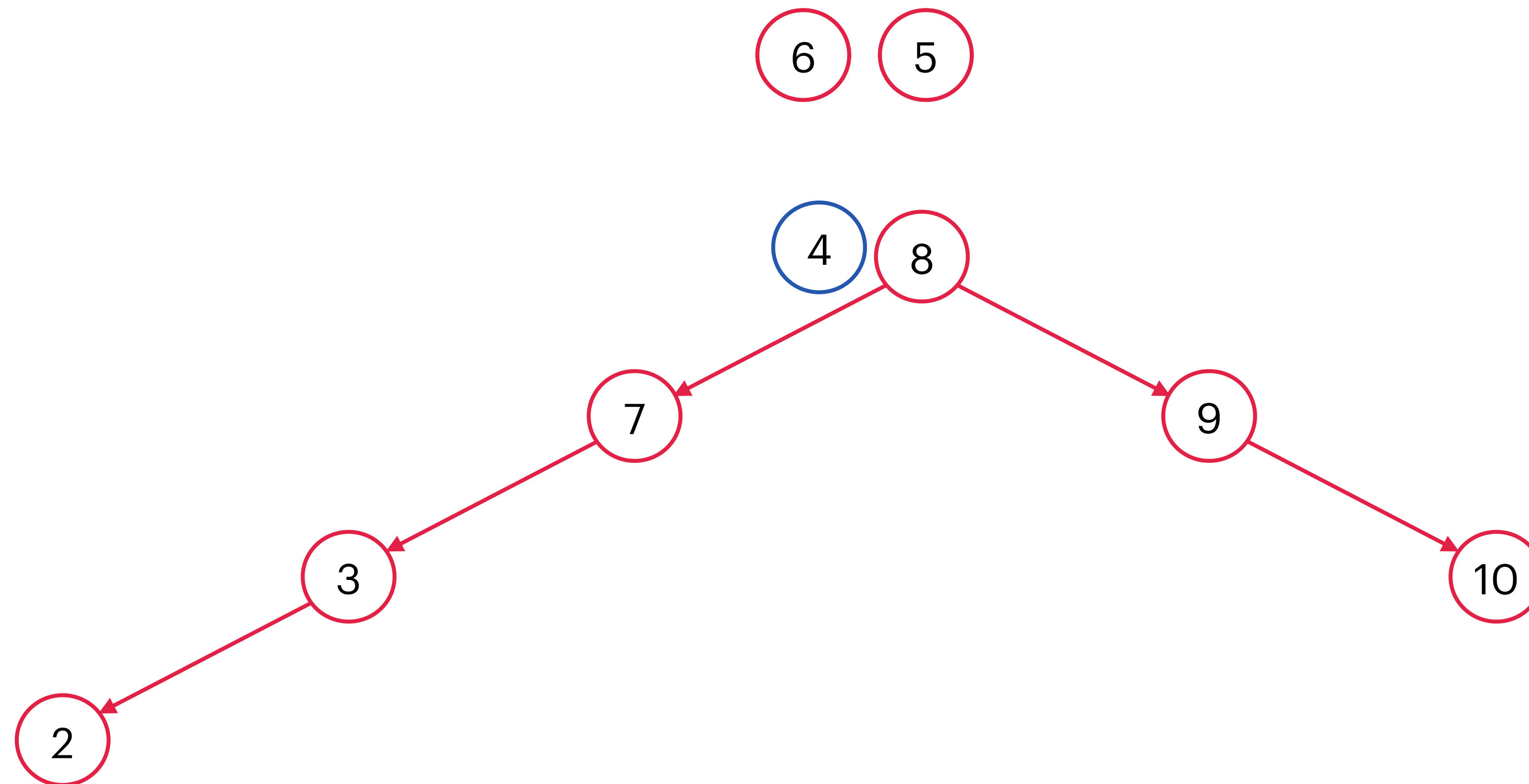
Arbori Binari Ordonati



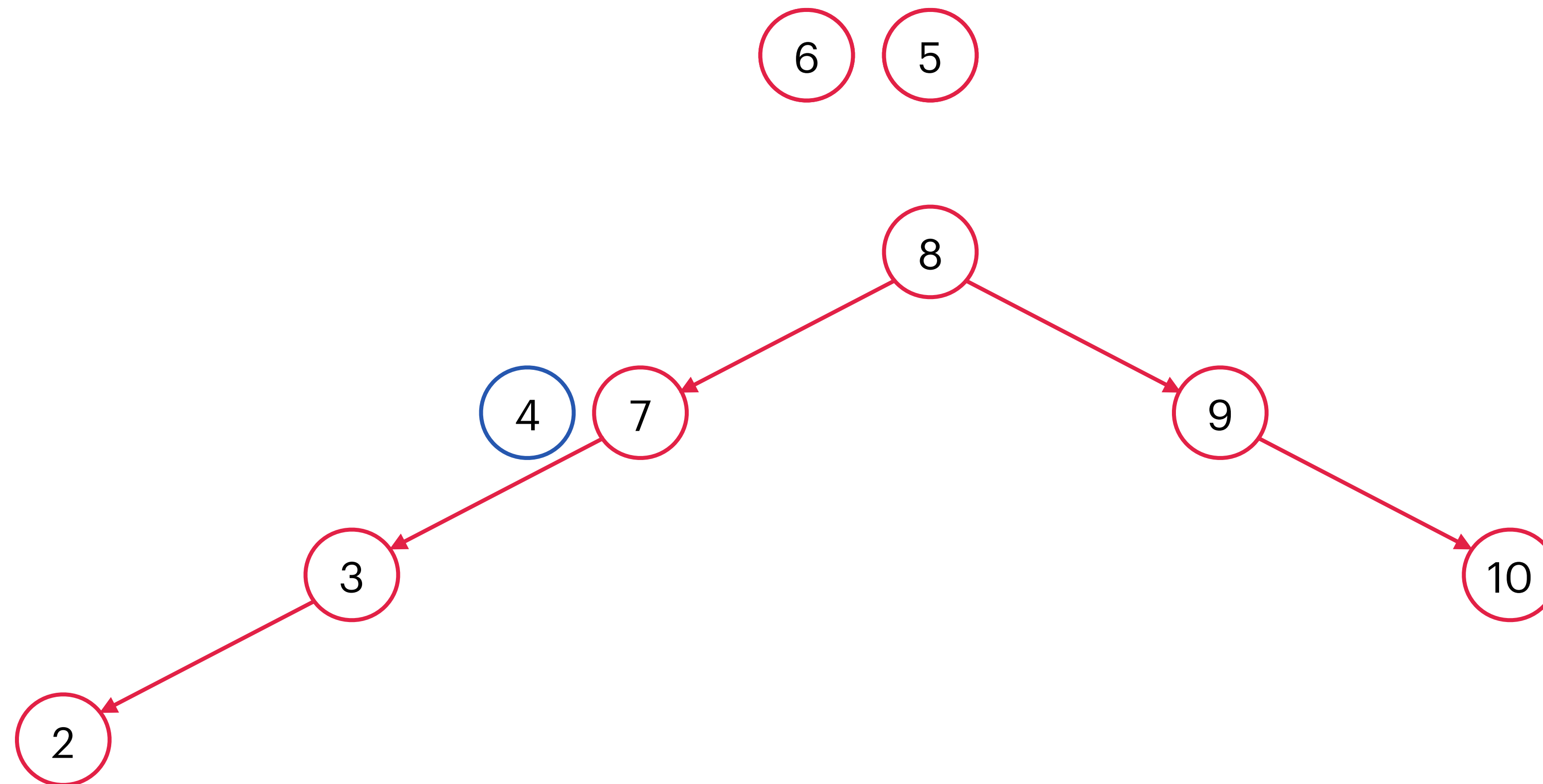
Arbori Binari Ordonati



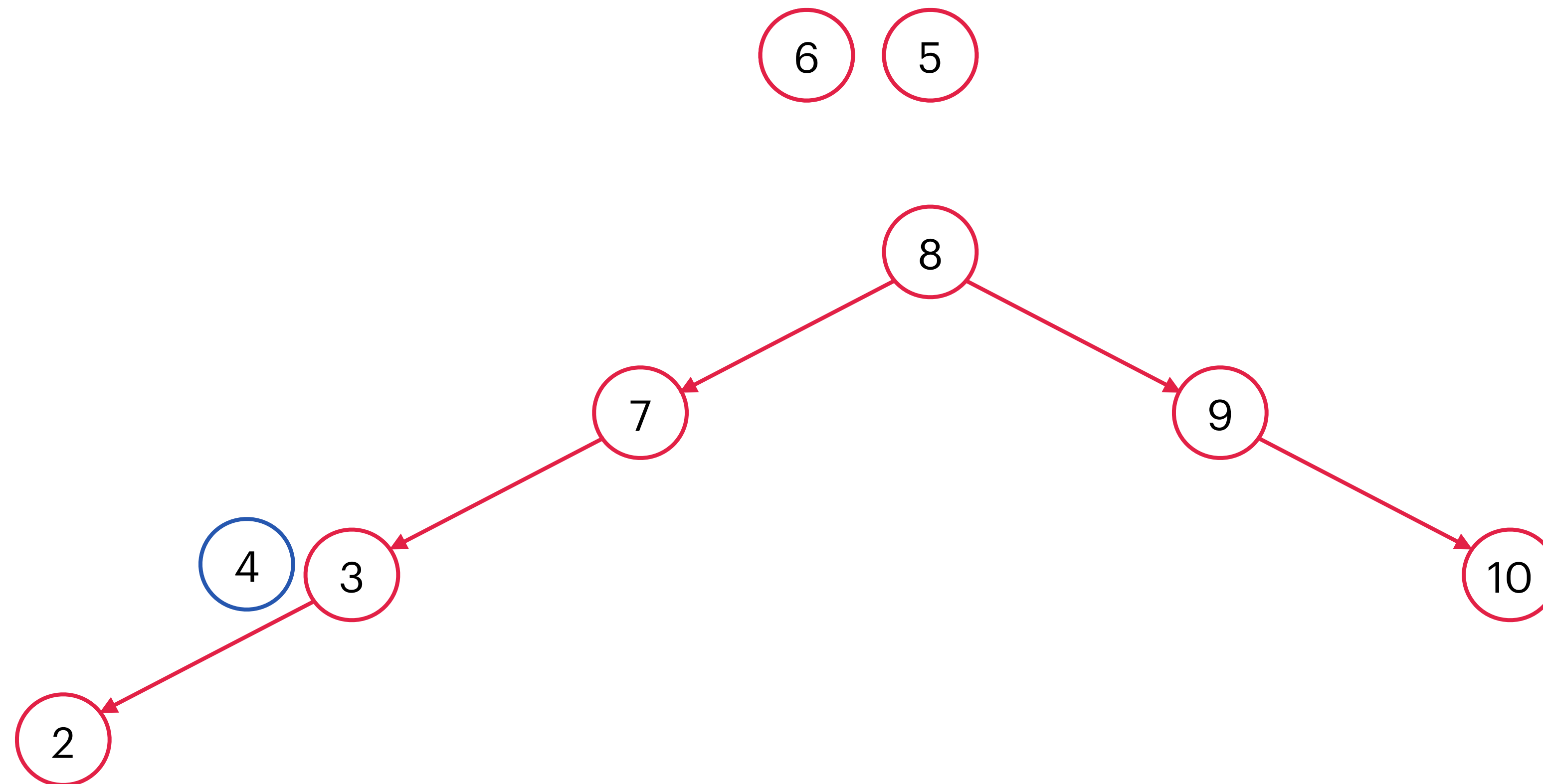
Arbori Binari Ordonati



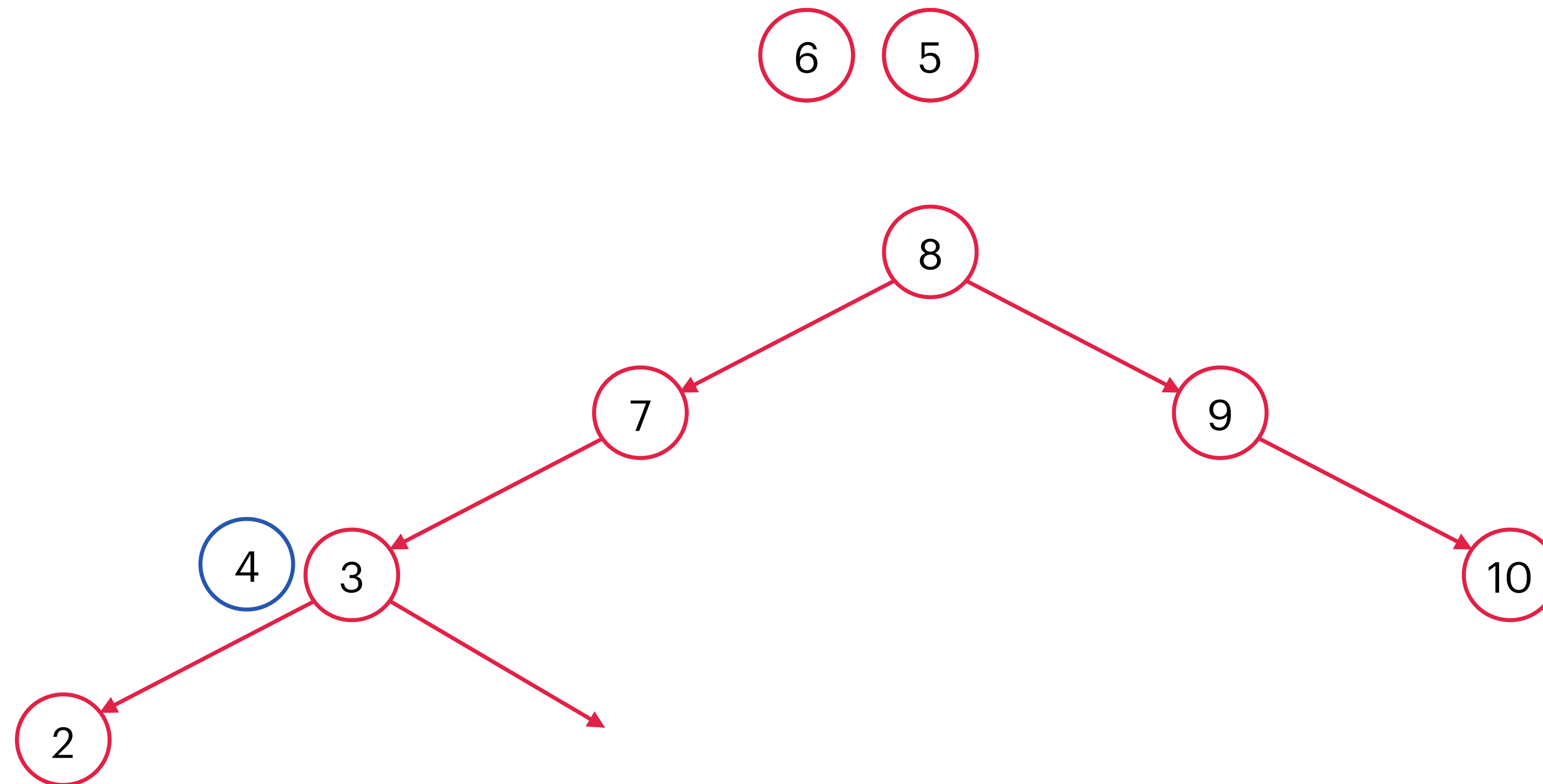
Arbori Binari Ordonati



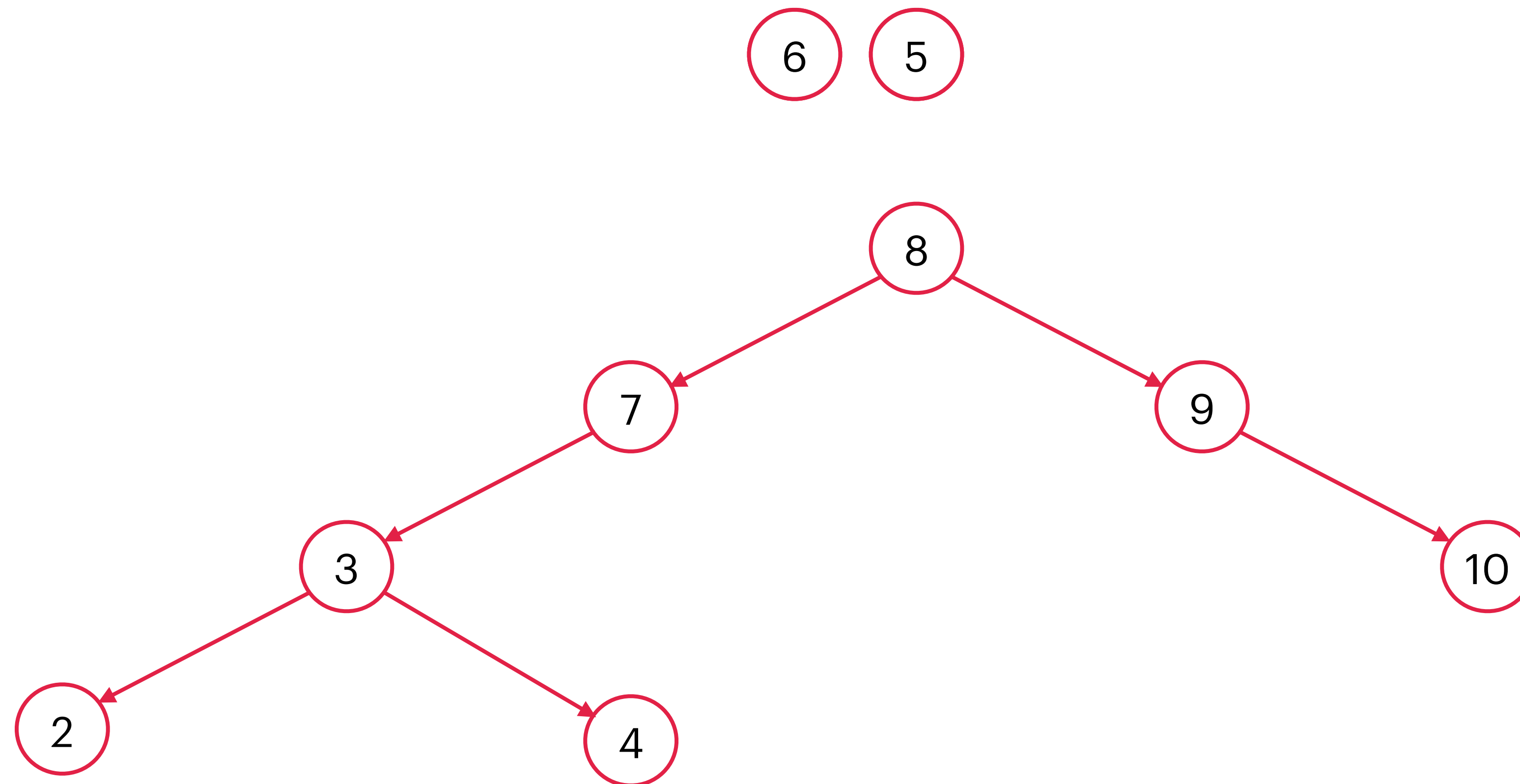
Arbori Binari Ordonati



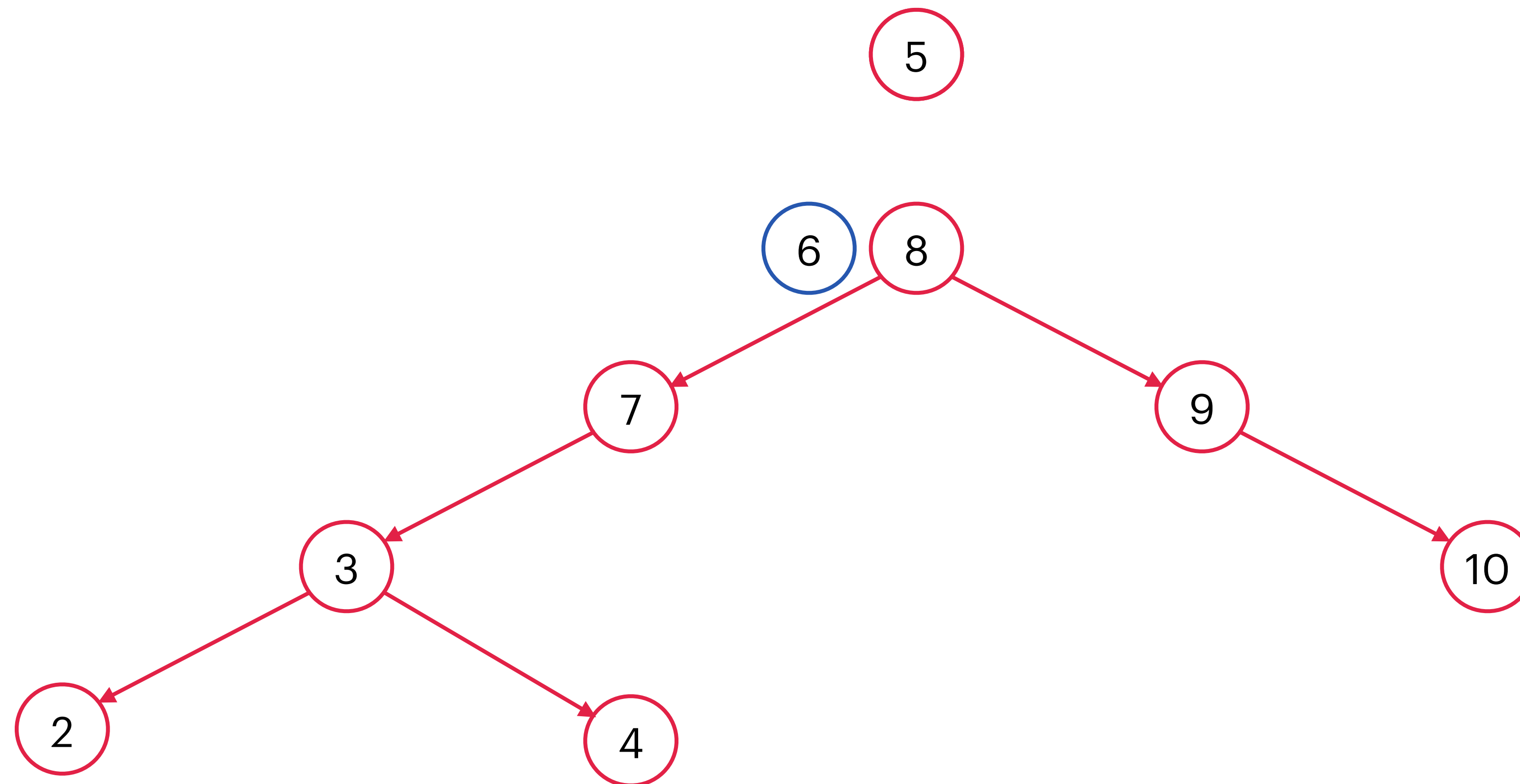
Arbori Binari Ordonati



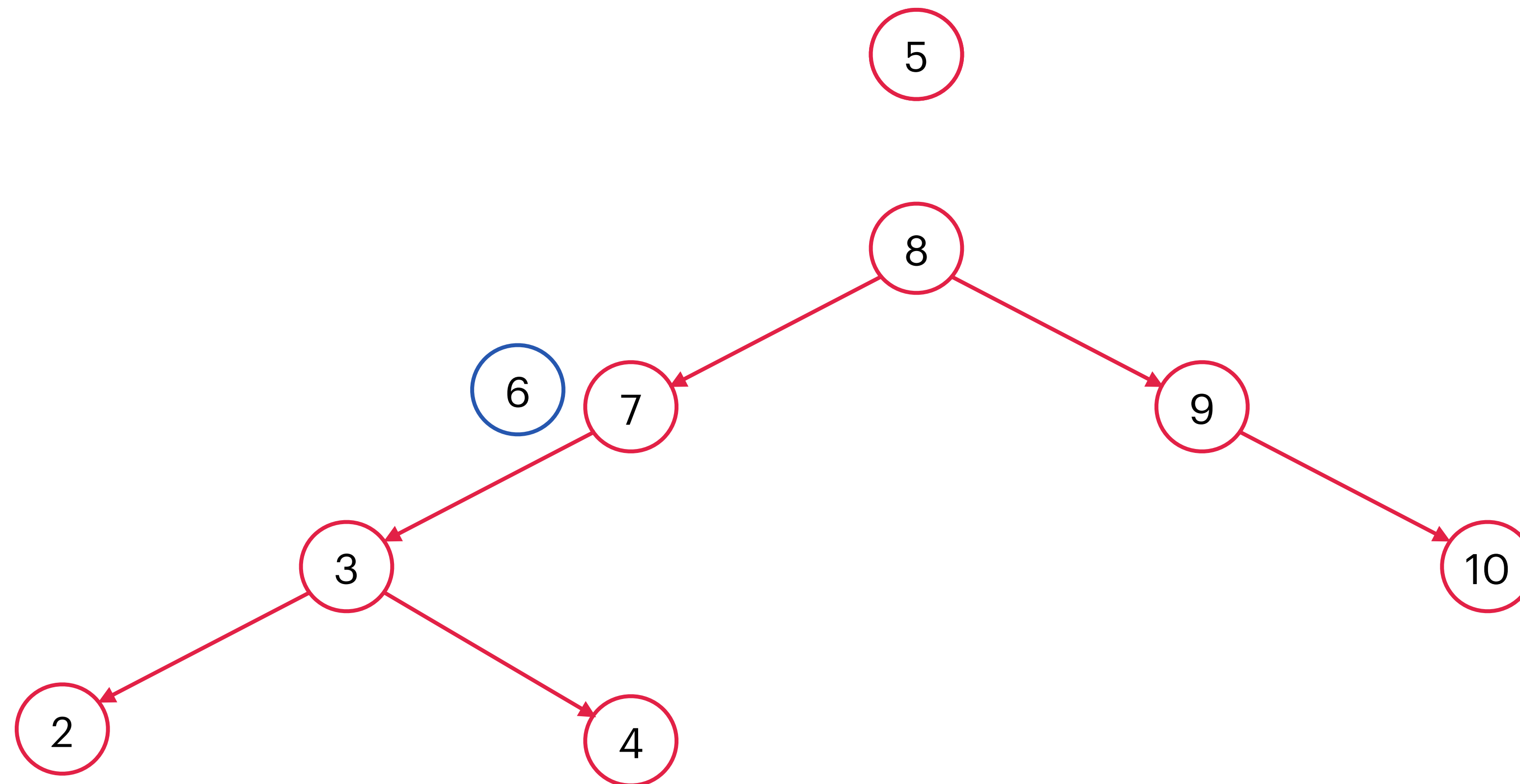
Arbori Binari Ordonati



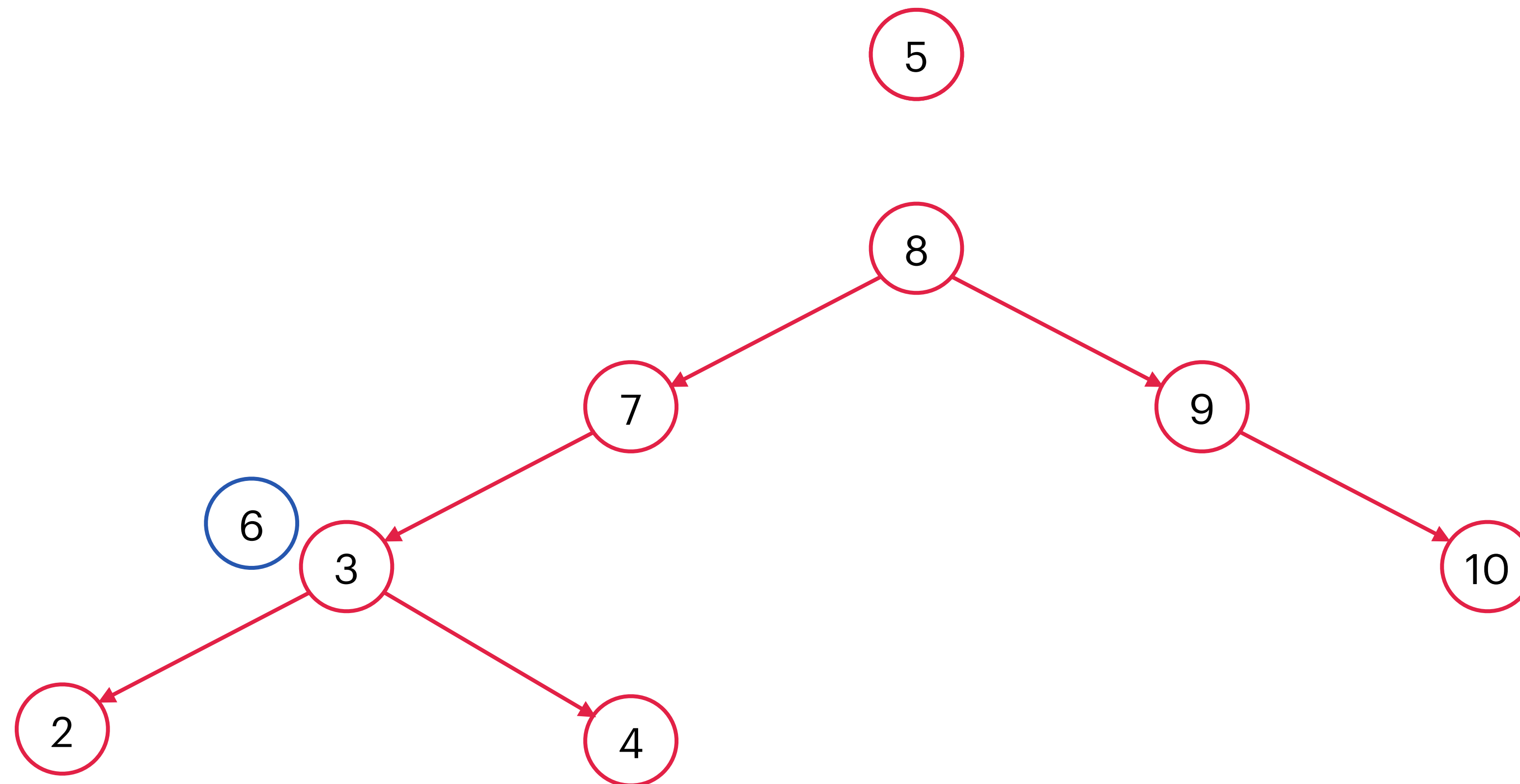
Arbori Binari Ordonati



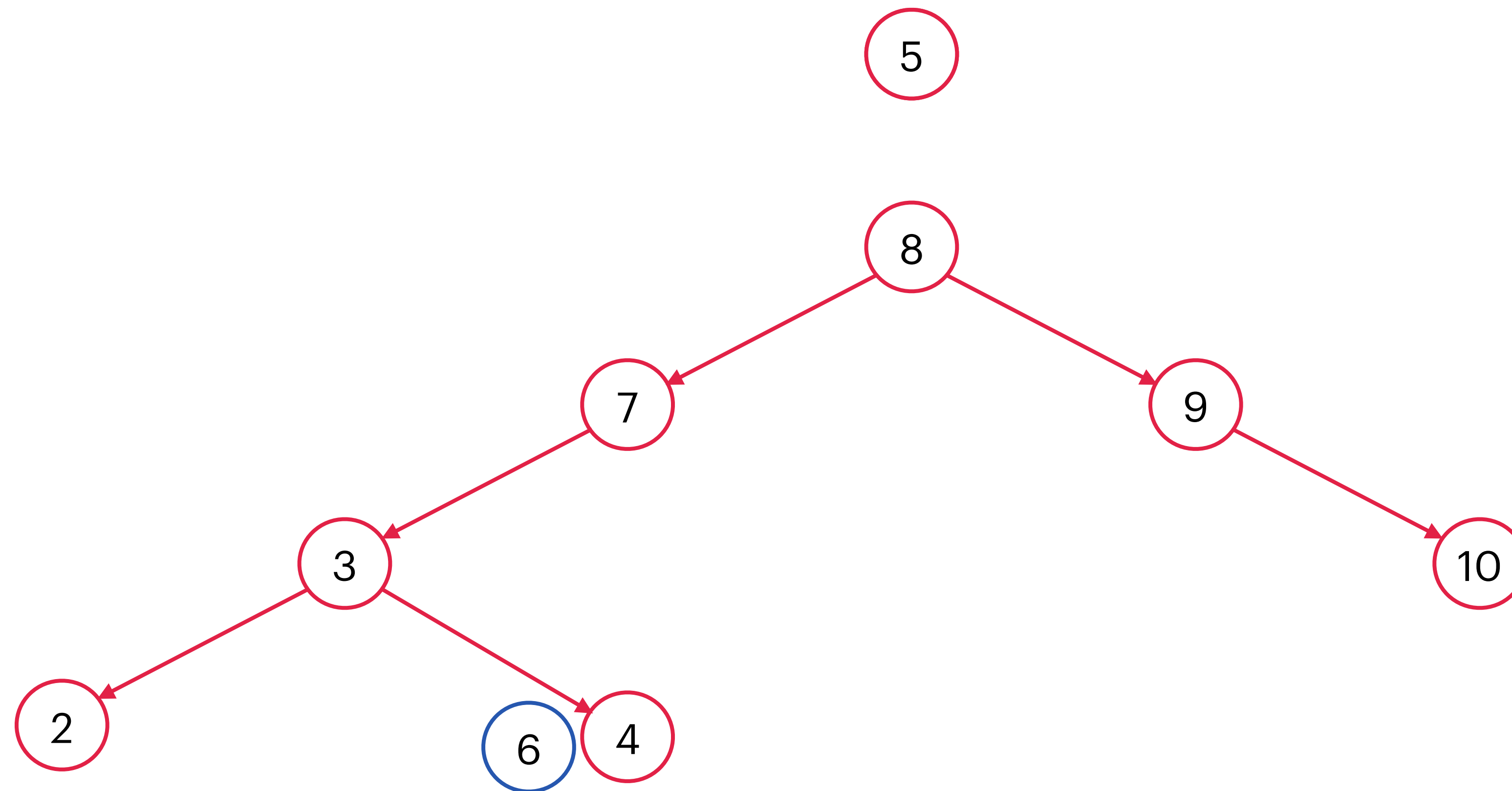
Arbori Binari Ordonati



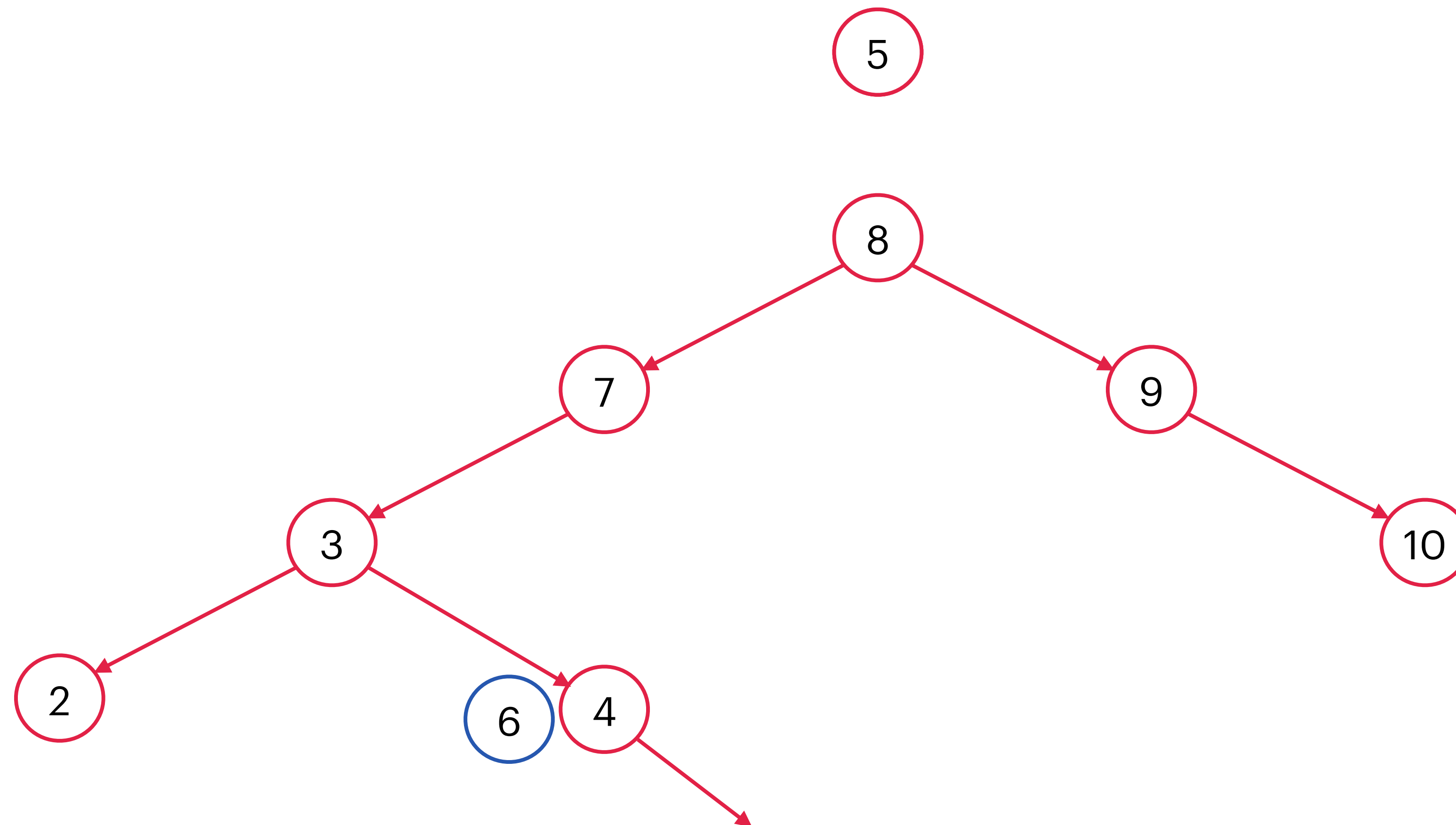
Arbori Binari Ordonati



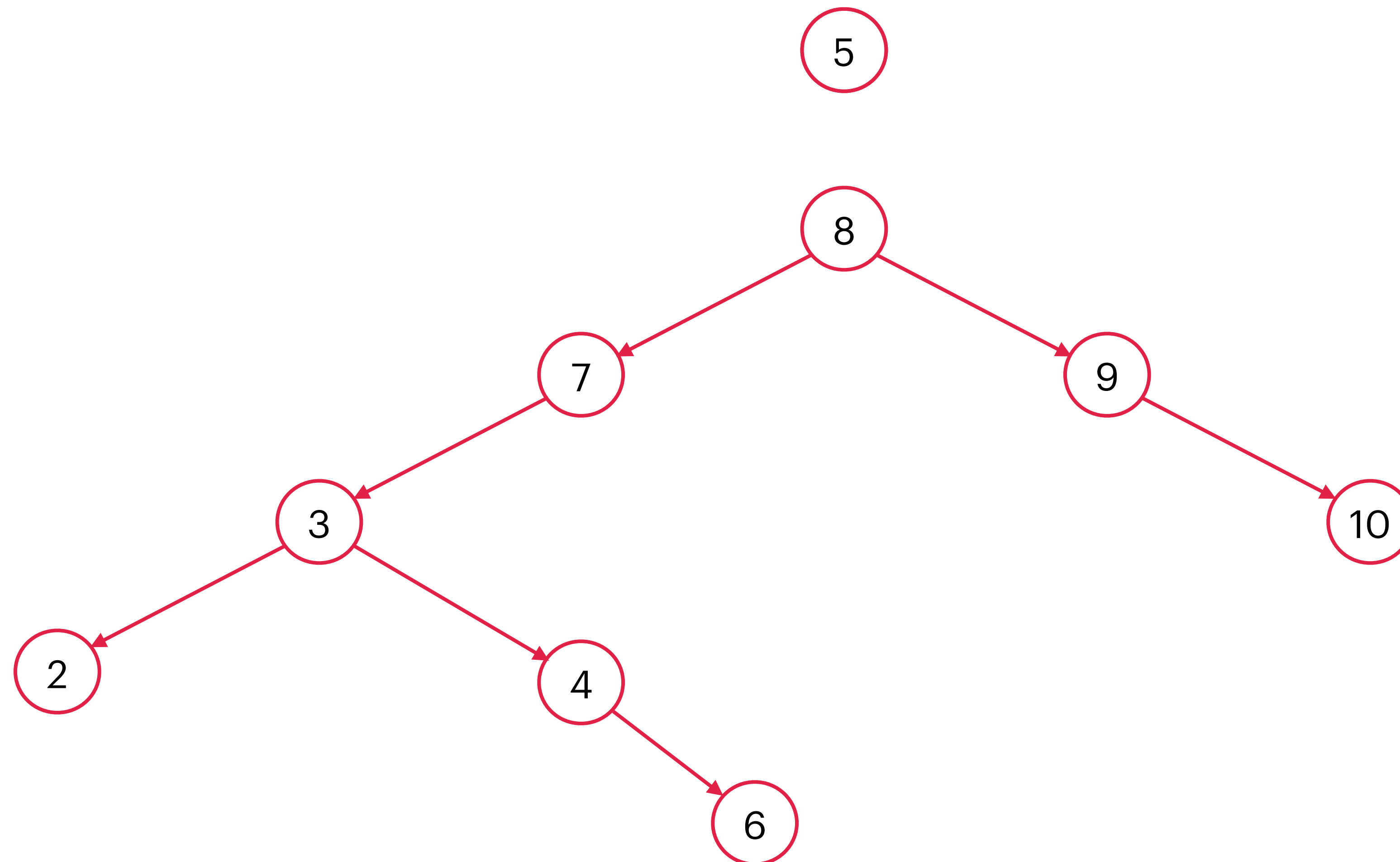
Arbori Binari Ordonati



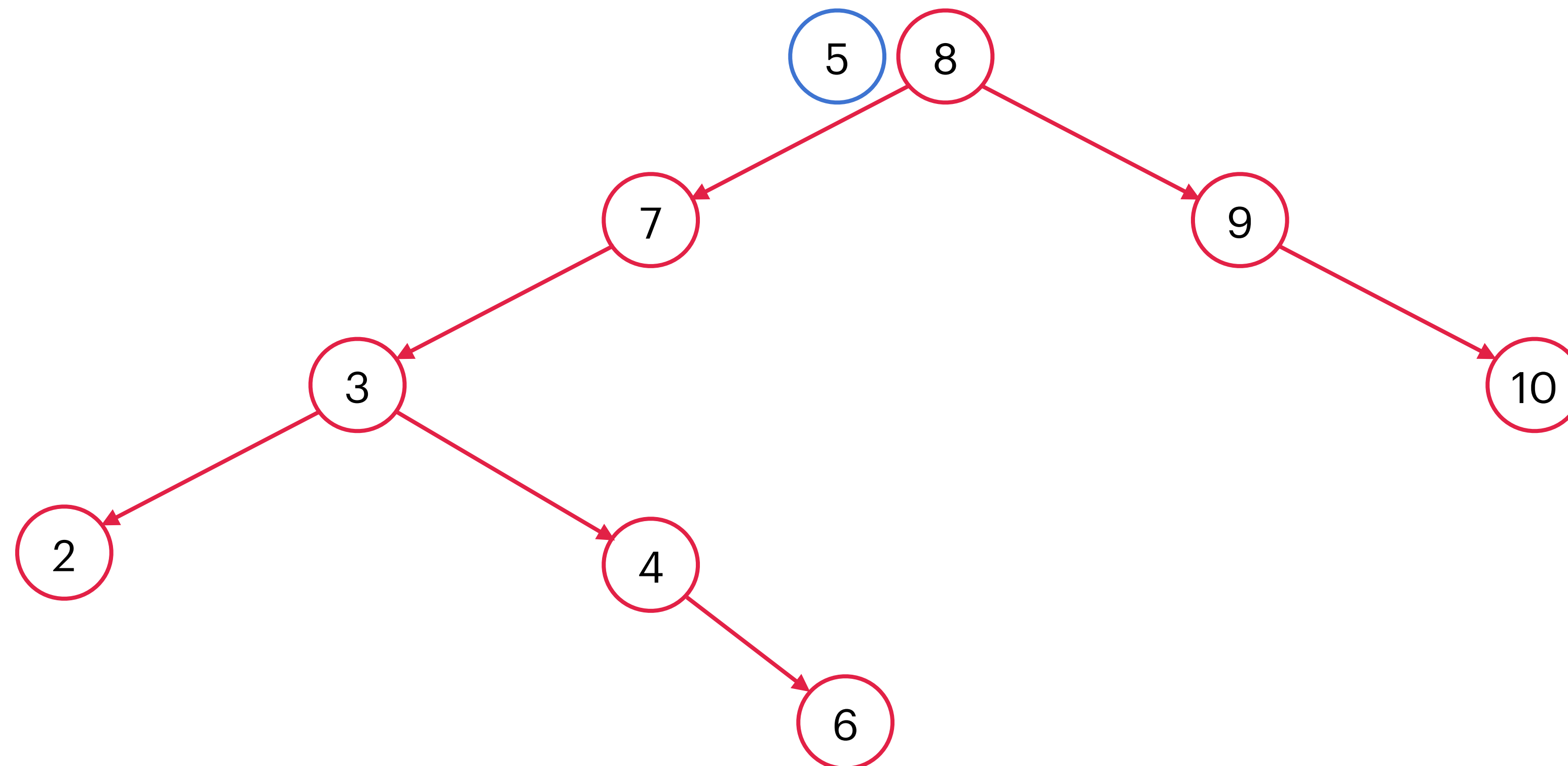
Arbori Binari Ordonati



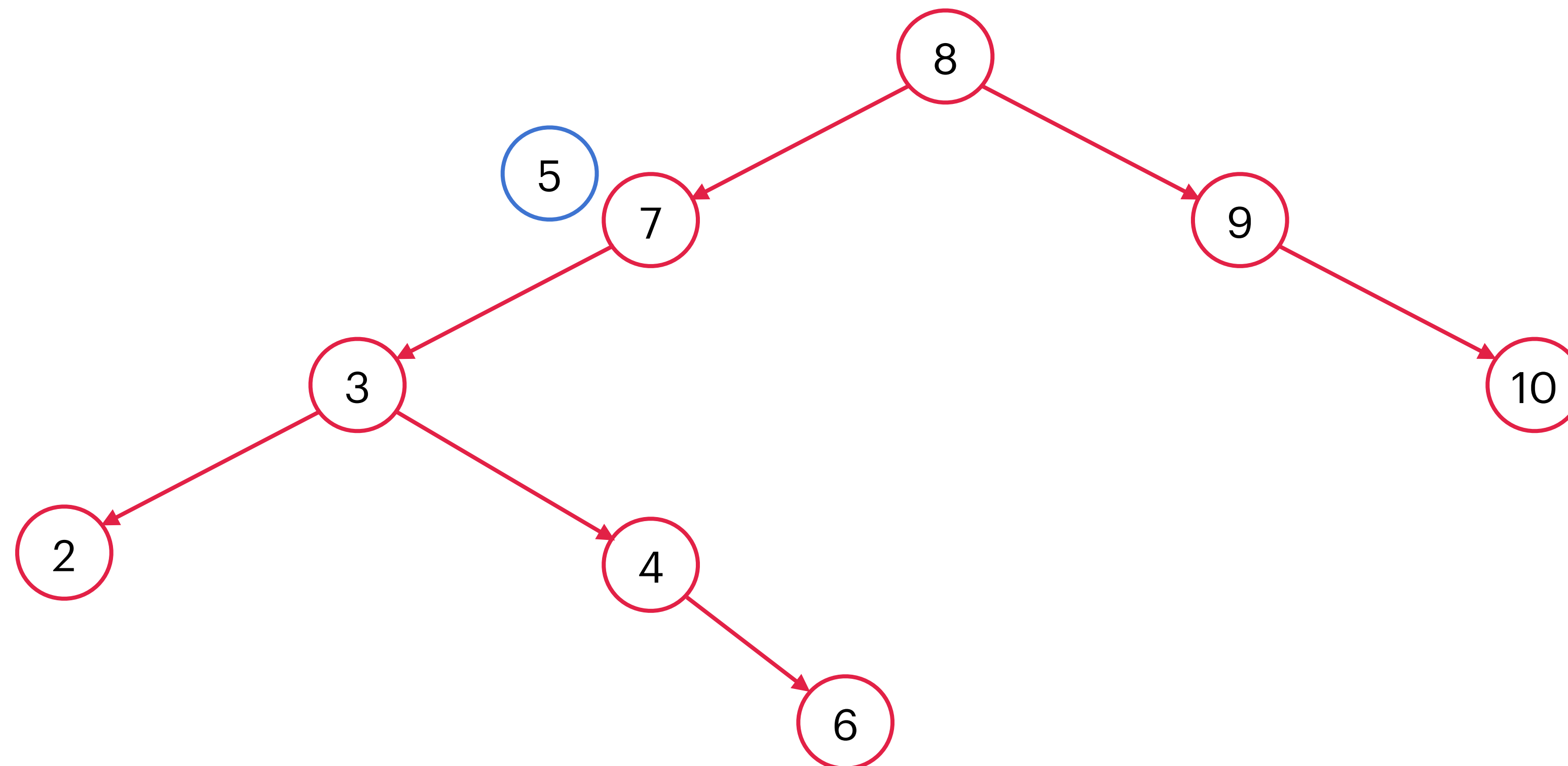
Arbori Binari Ordonati



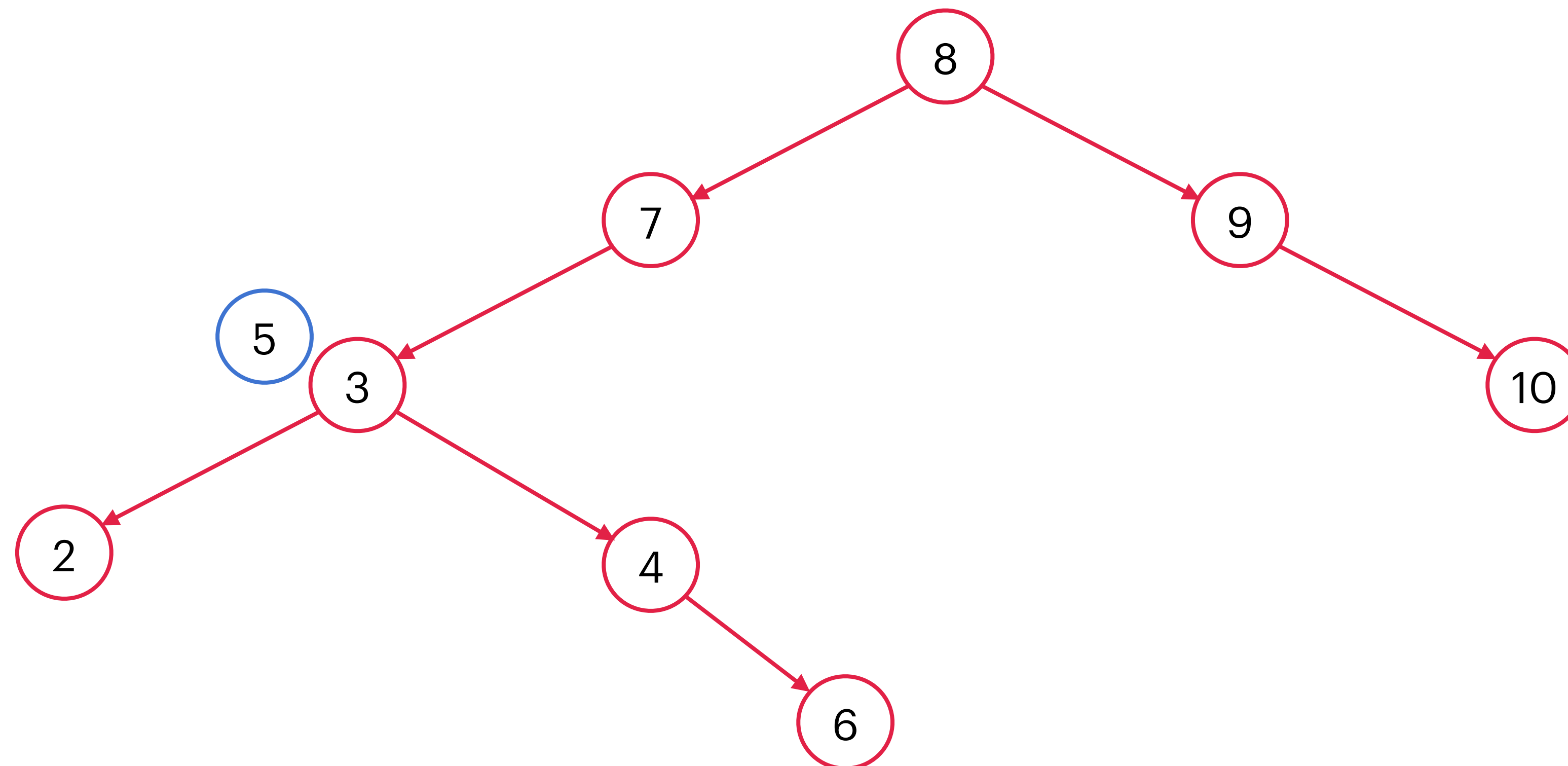
Arbori Binari Ordonati



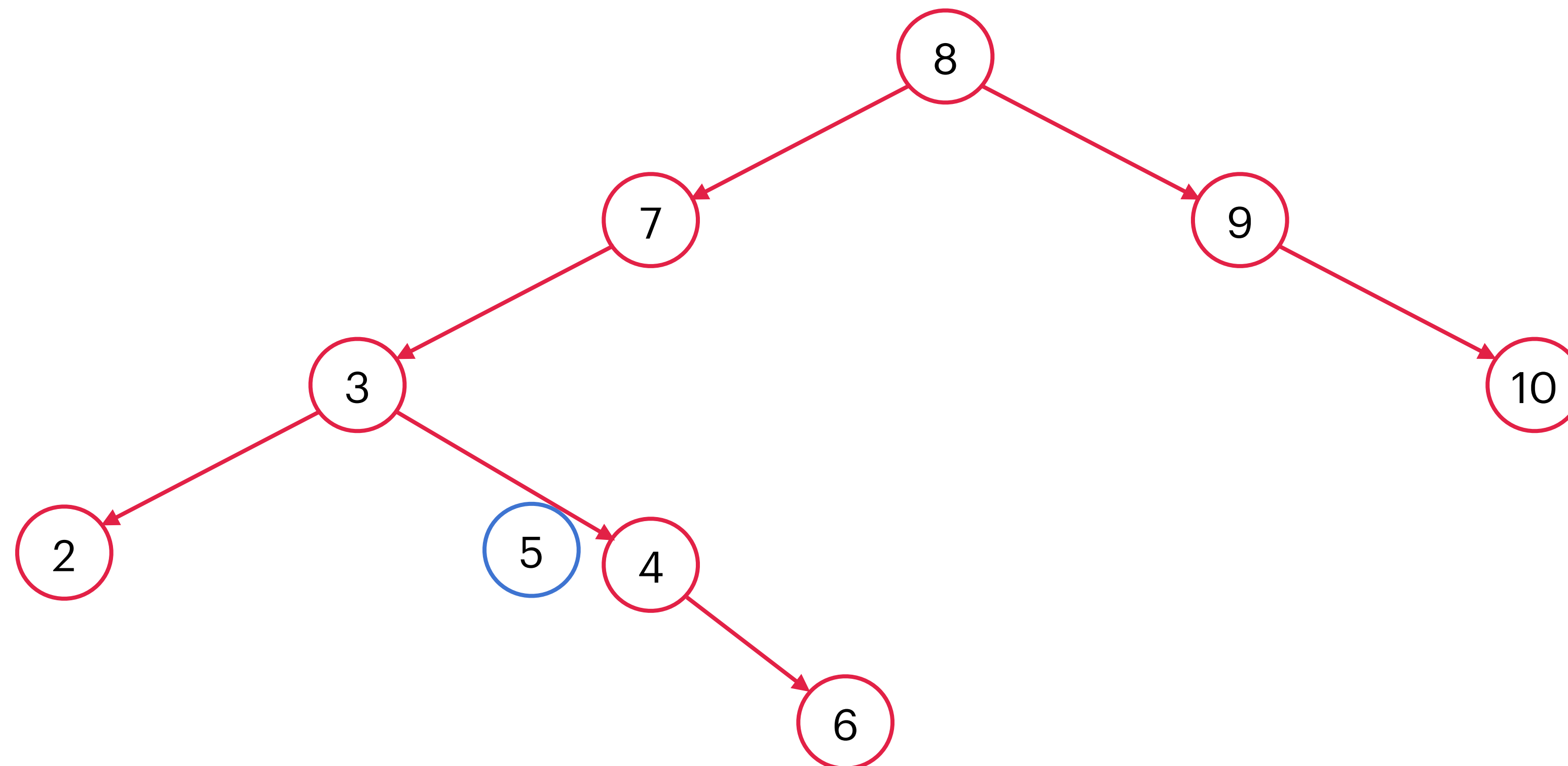
Arbori Binari Ordonati



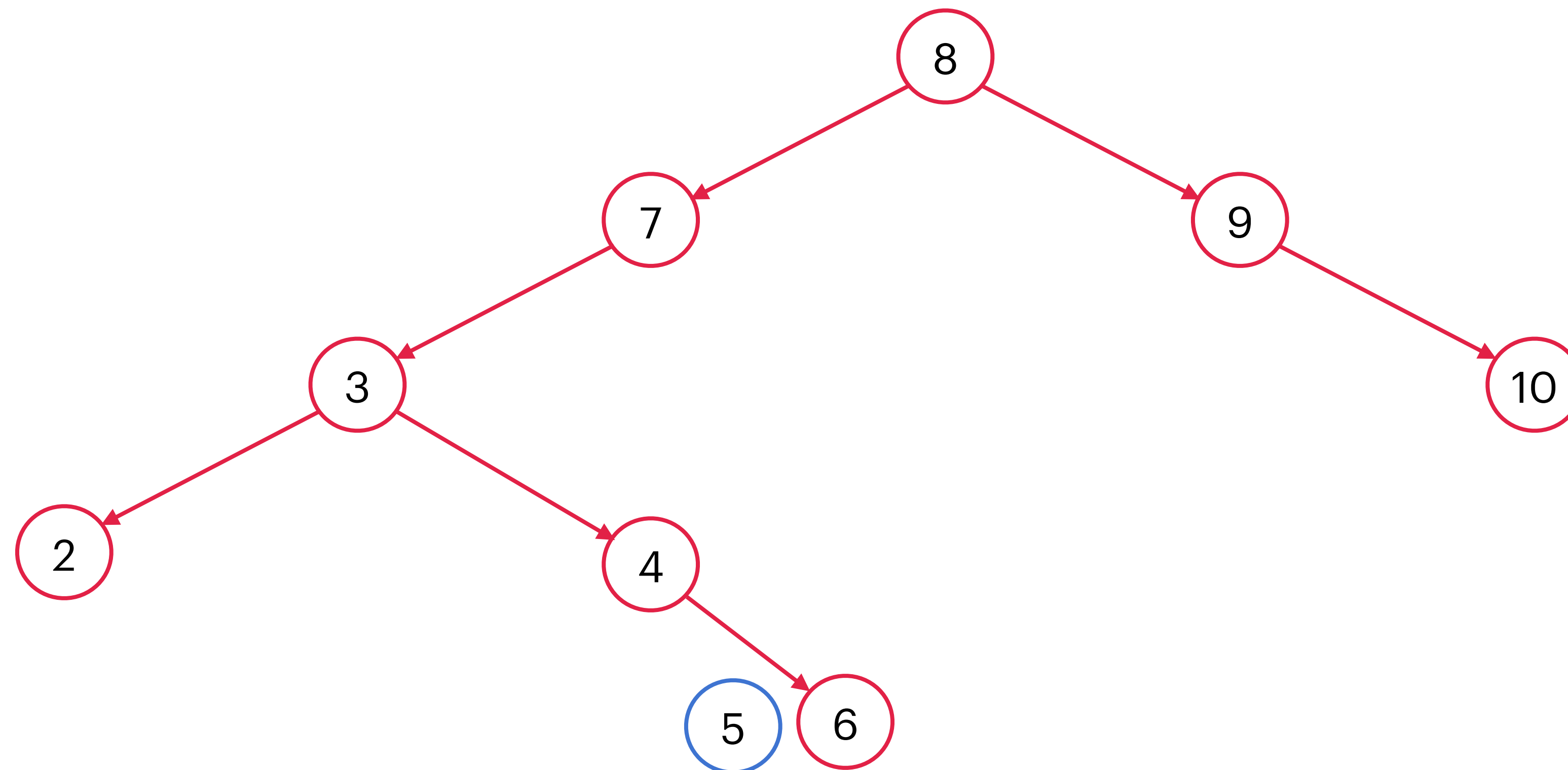
Arbori Binari Ordonati



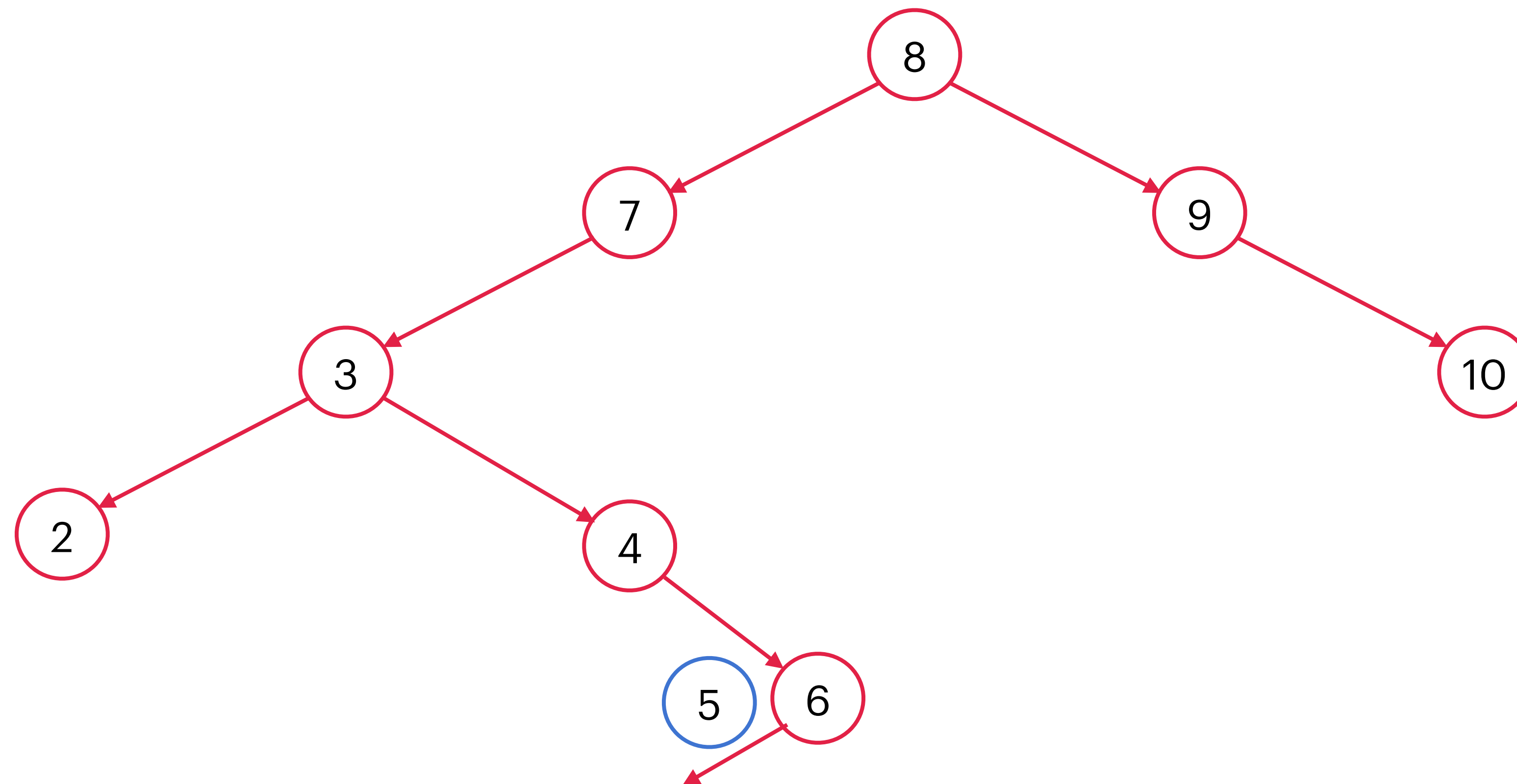
Arbori Binari Ordonati



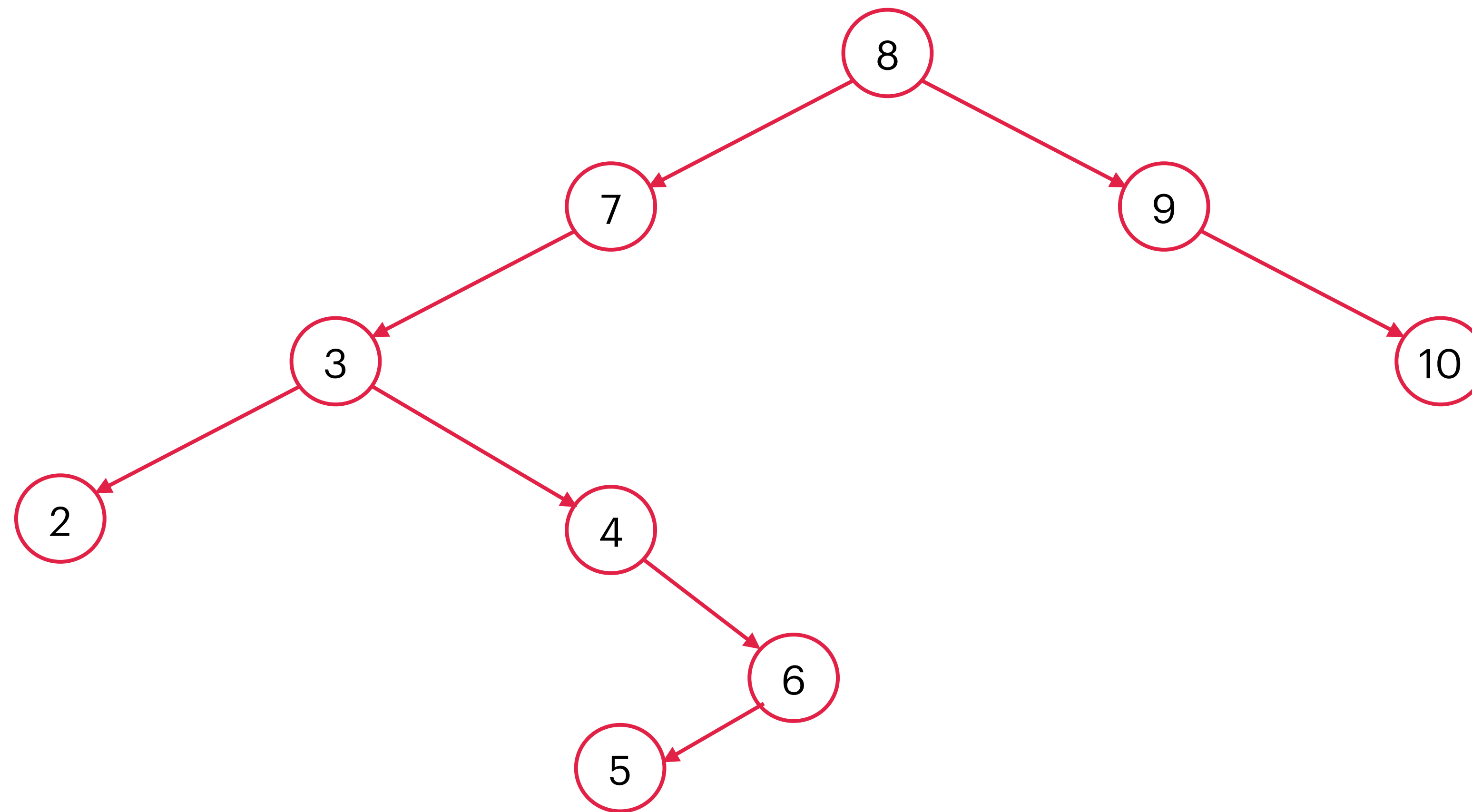
Arbori Binari Ordonati



Arbori Binari Ordonati



Arbori Binari Ordonati



Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există

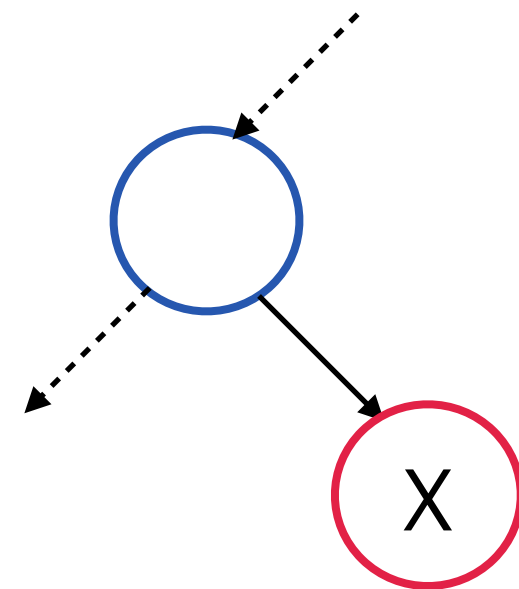
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



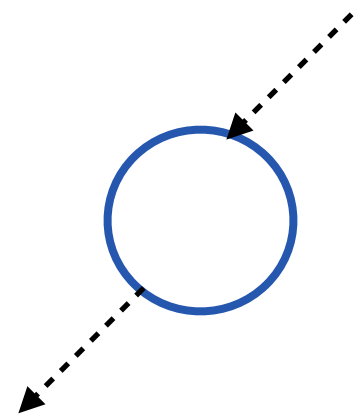
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



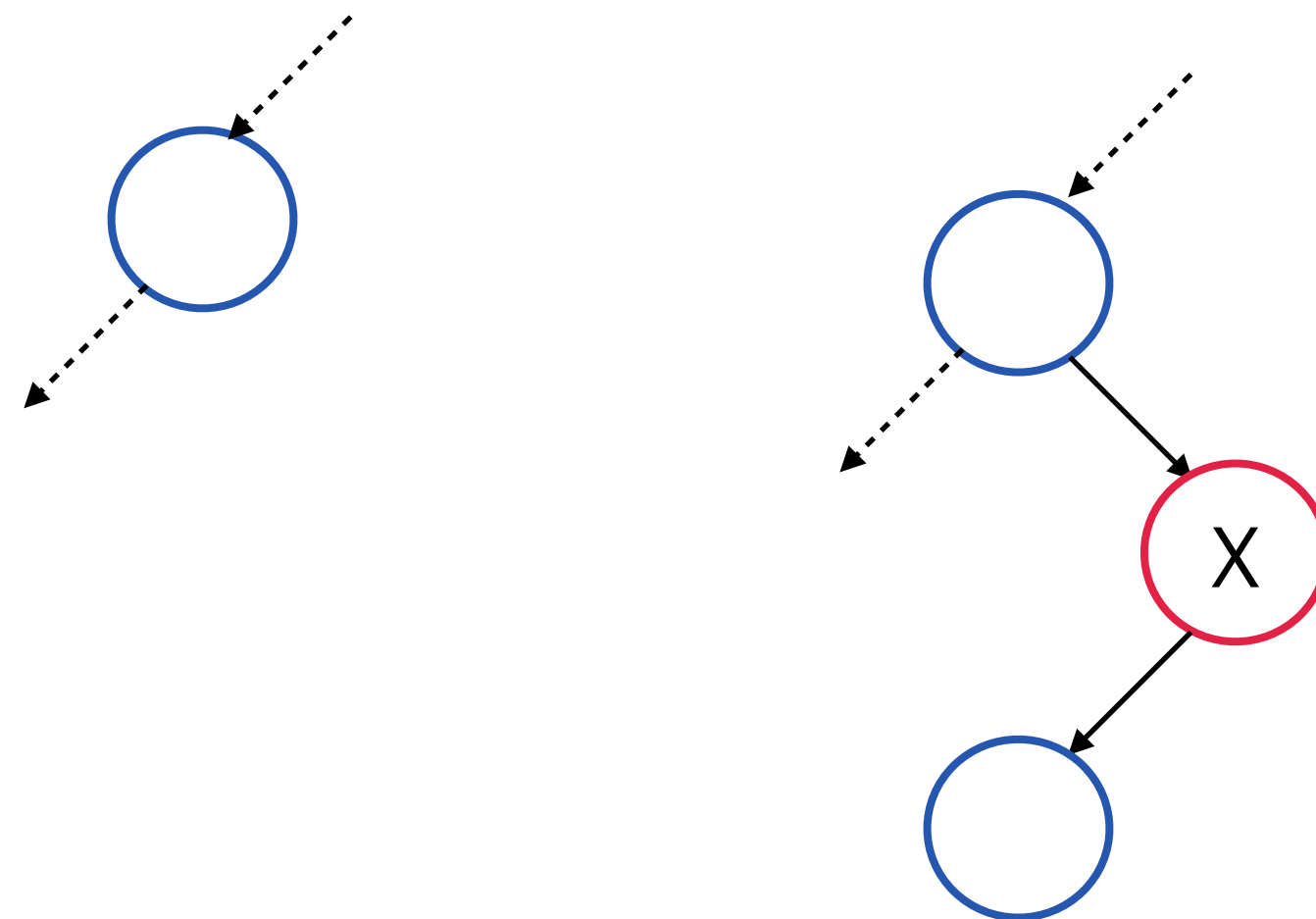
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



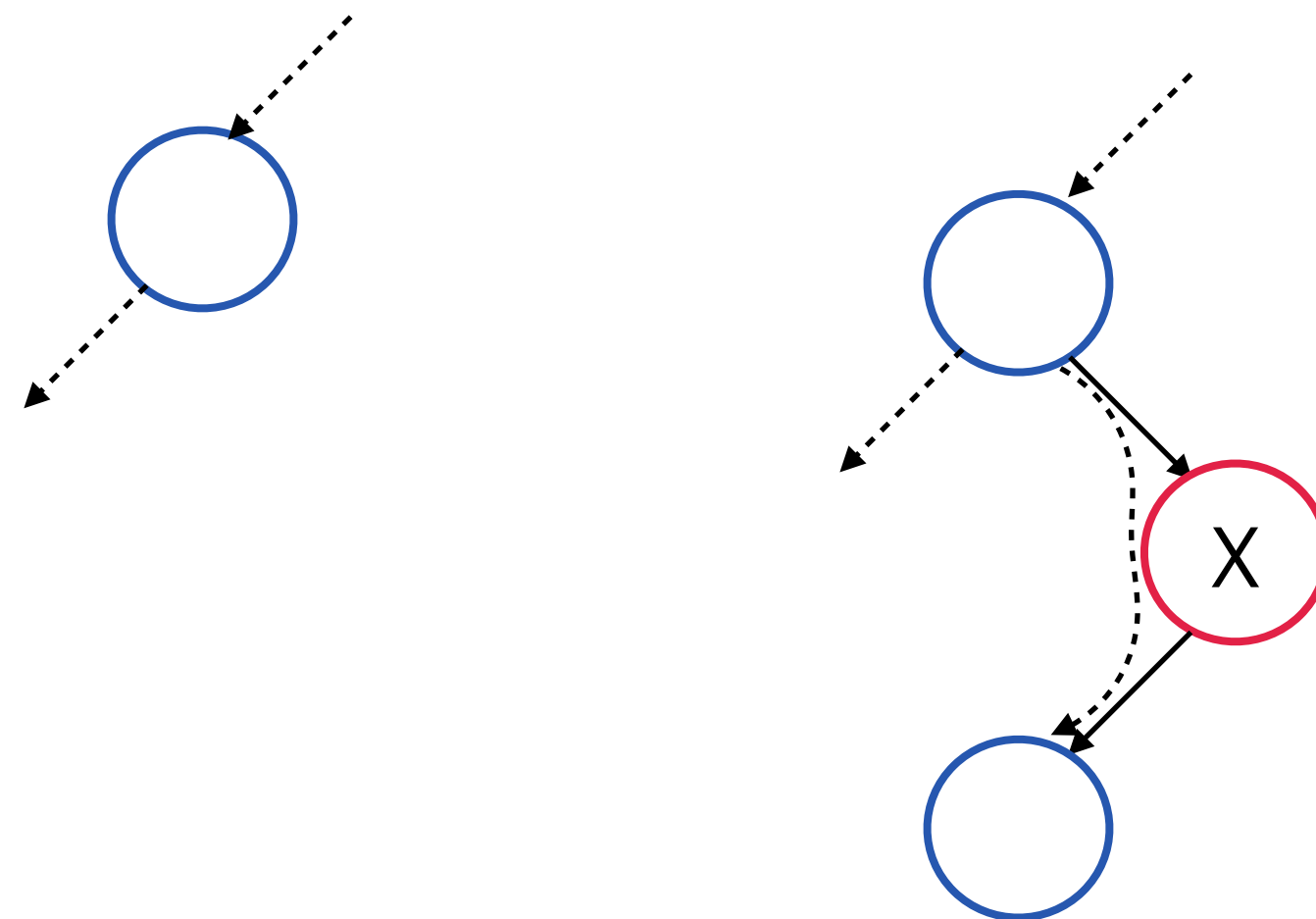
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



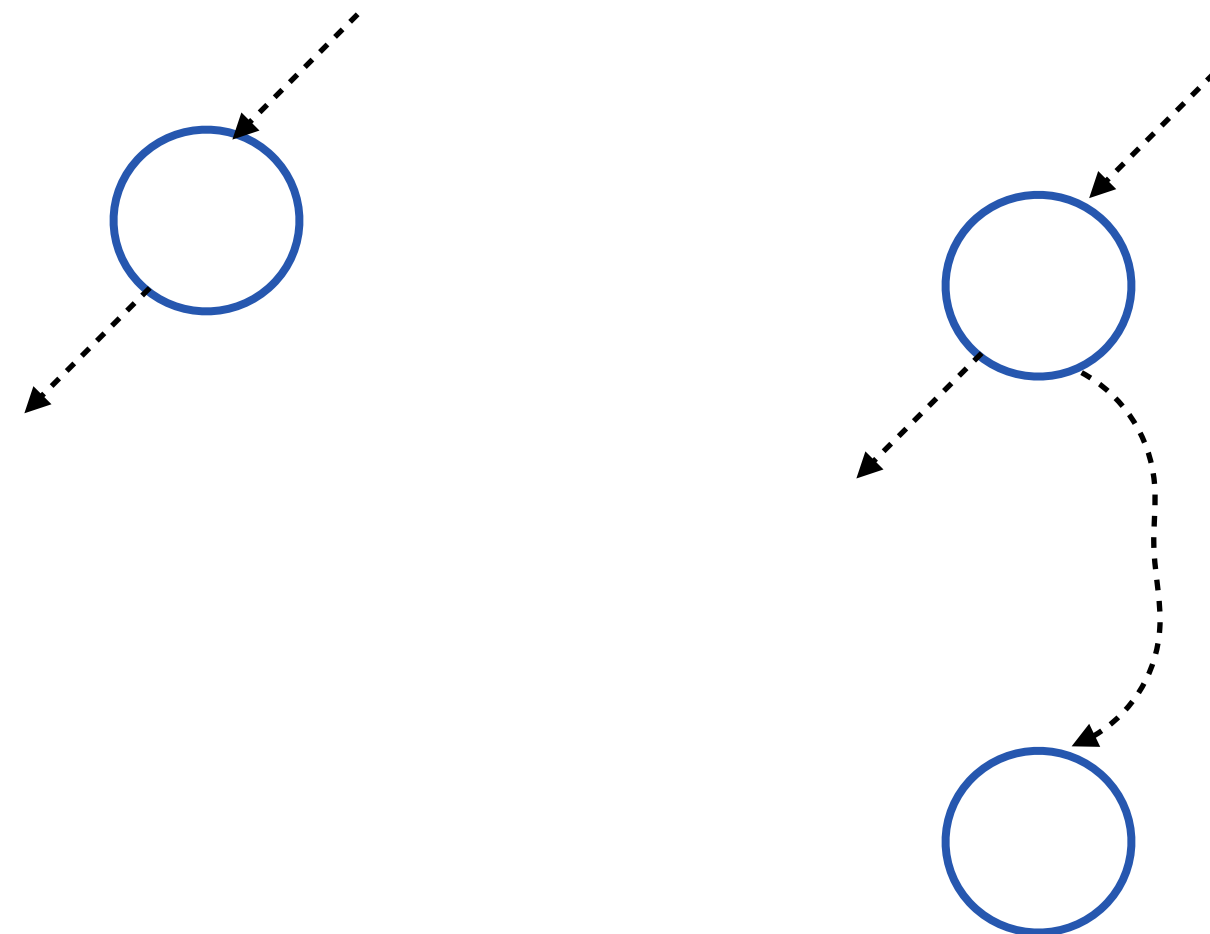
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



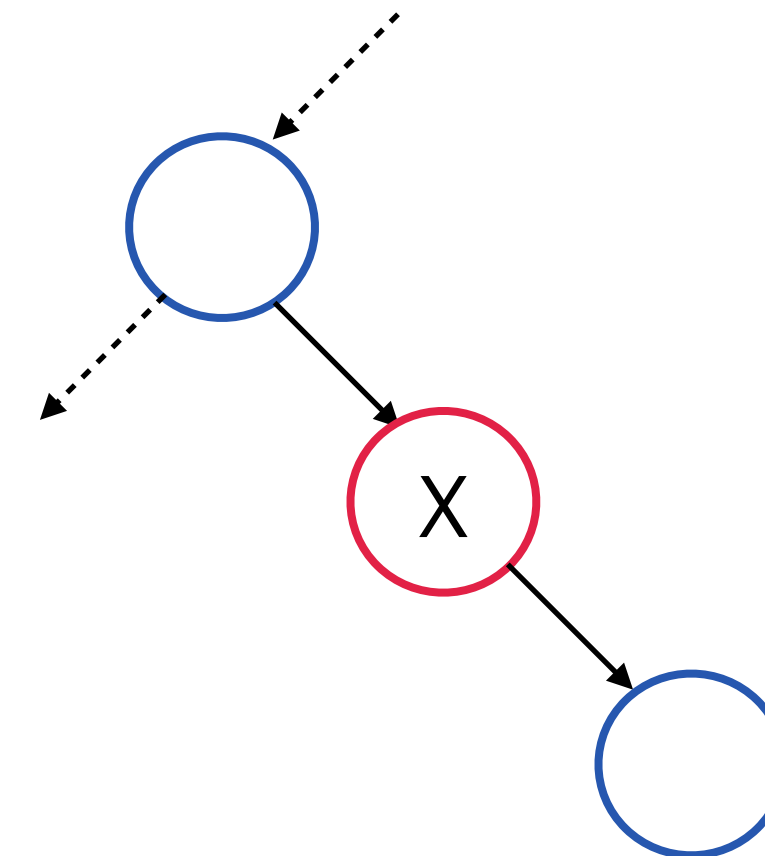
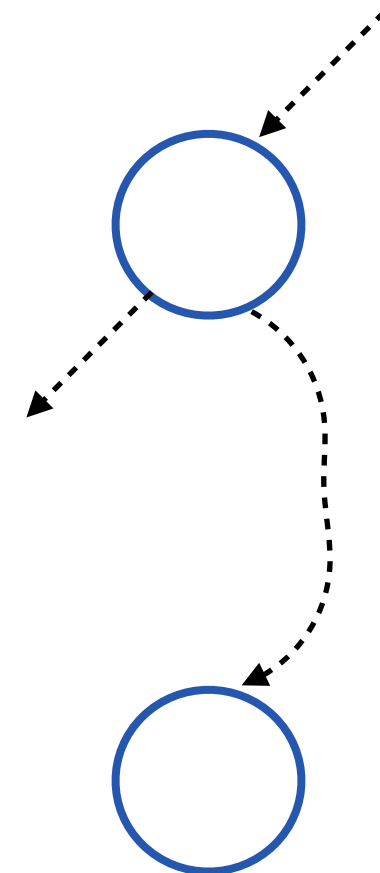
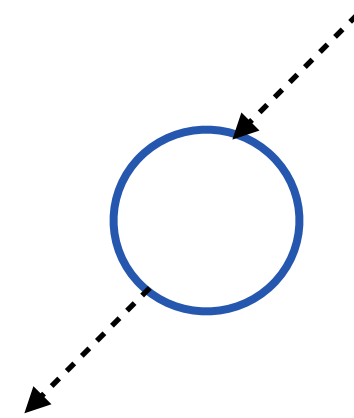
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



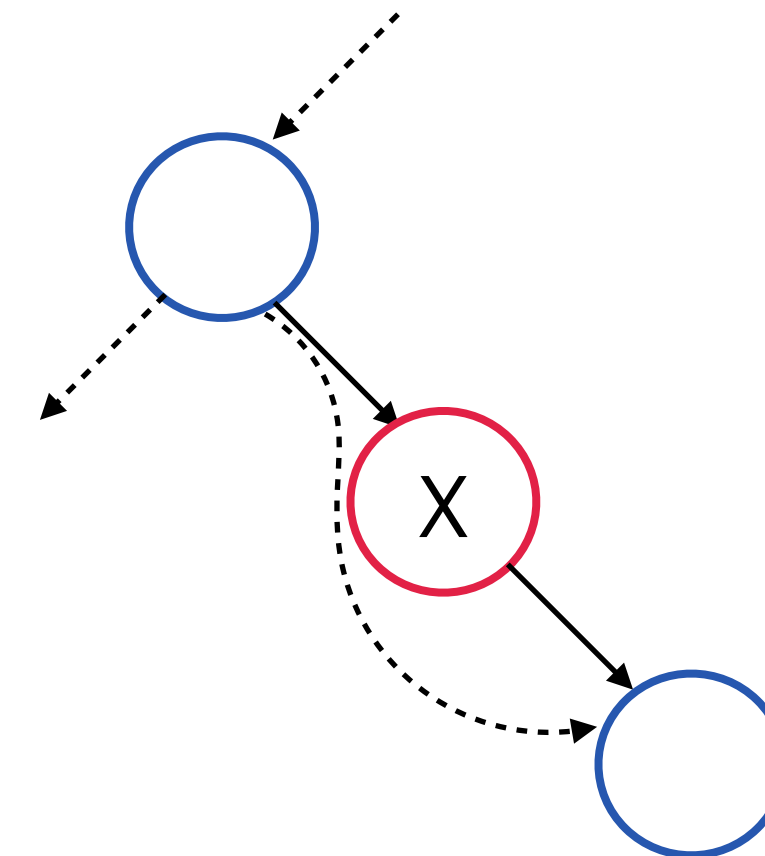
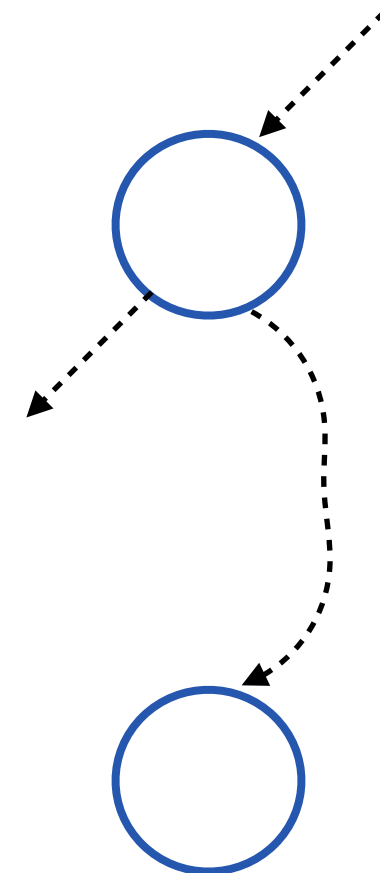
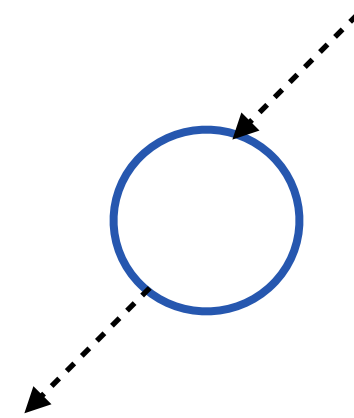
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



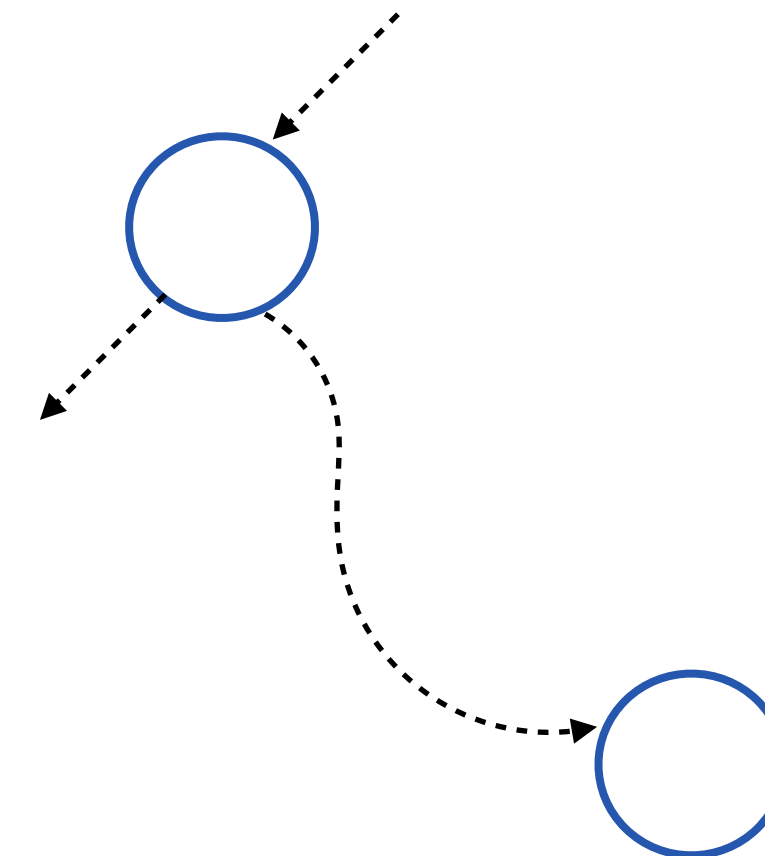
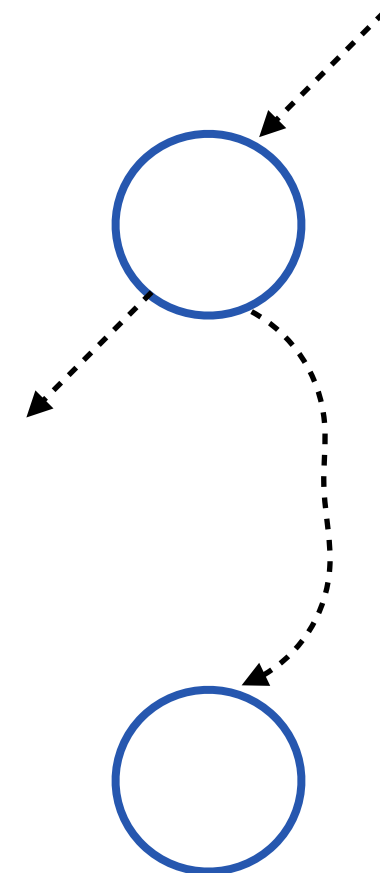
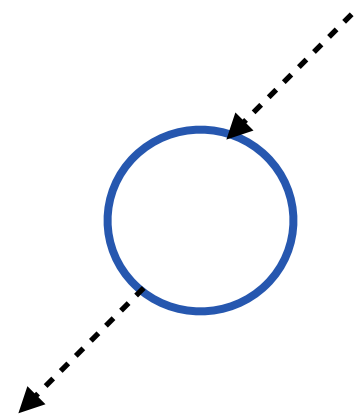
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



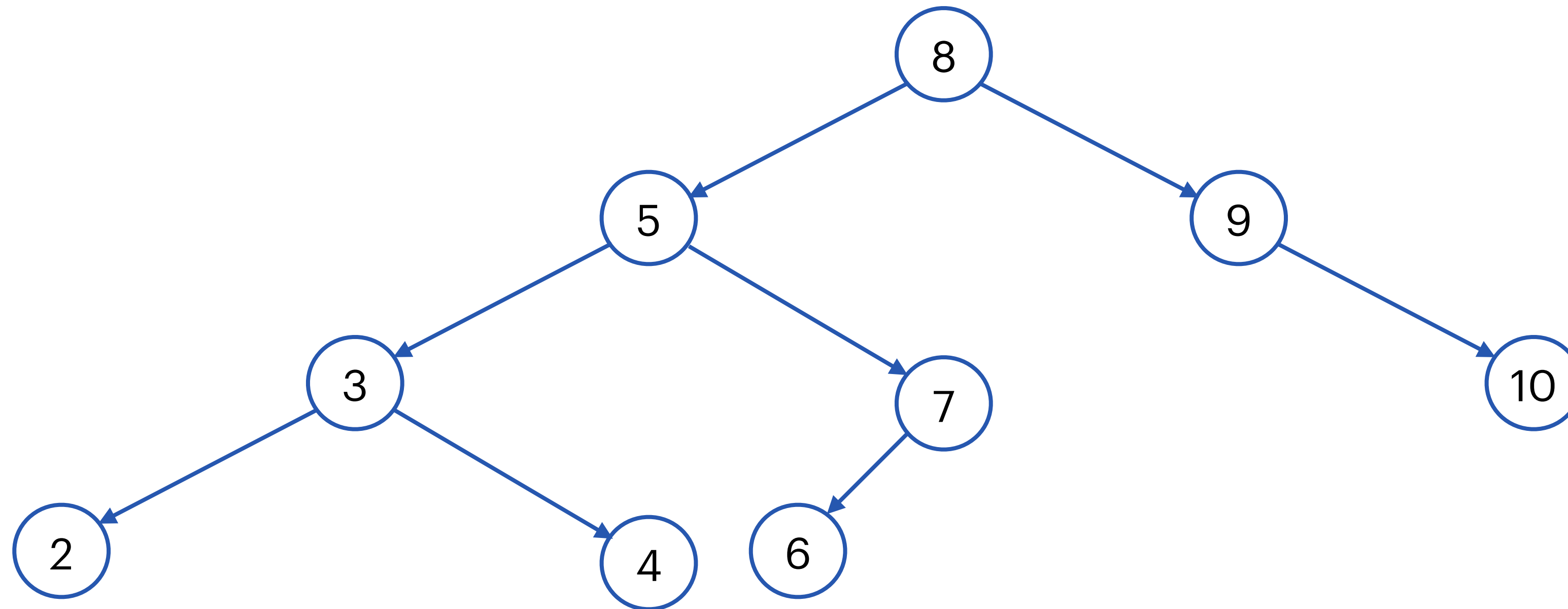
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



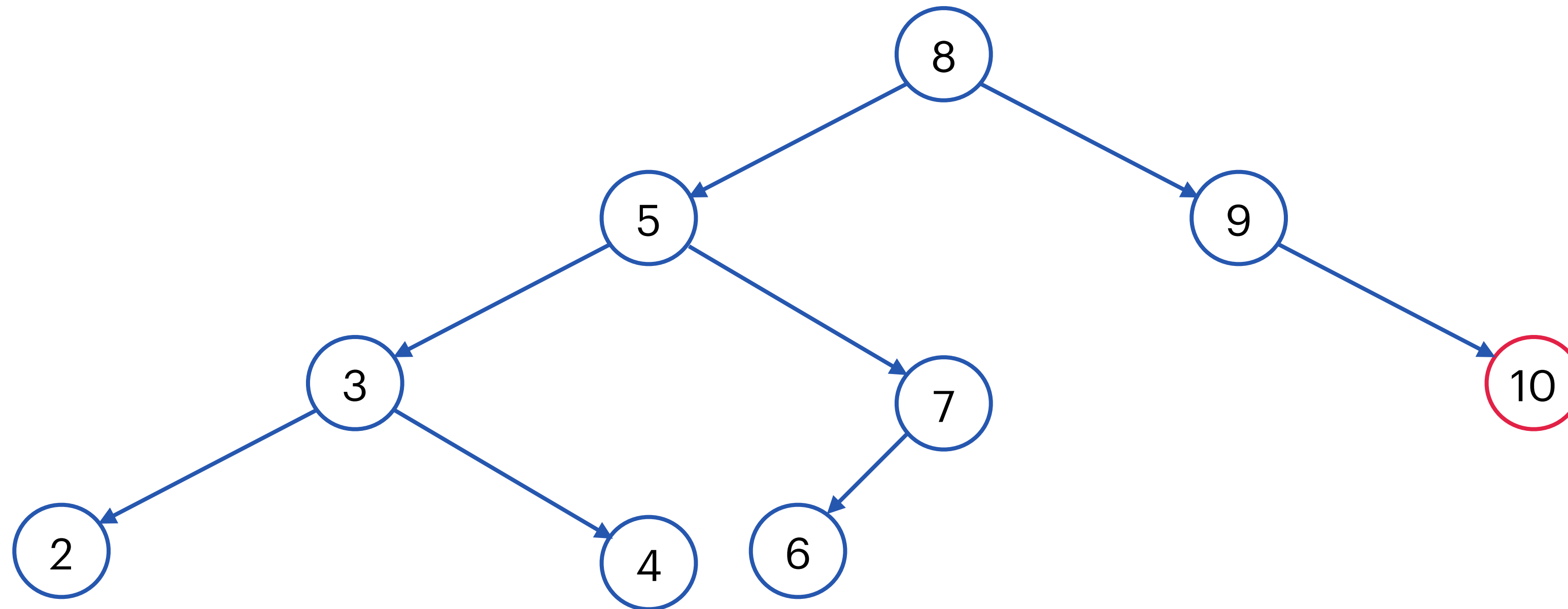
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



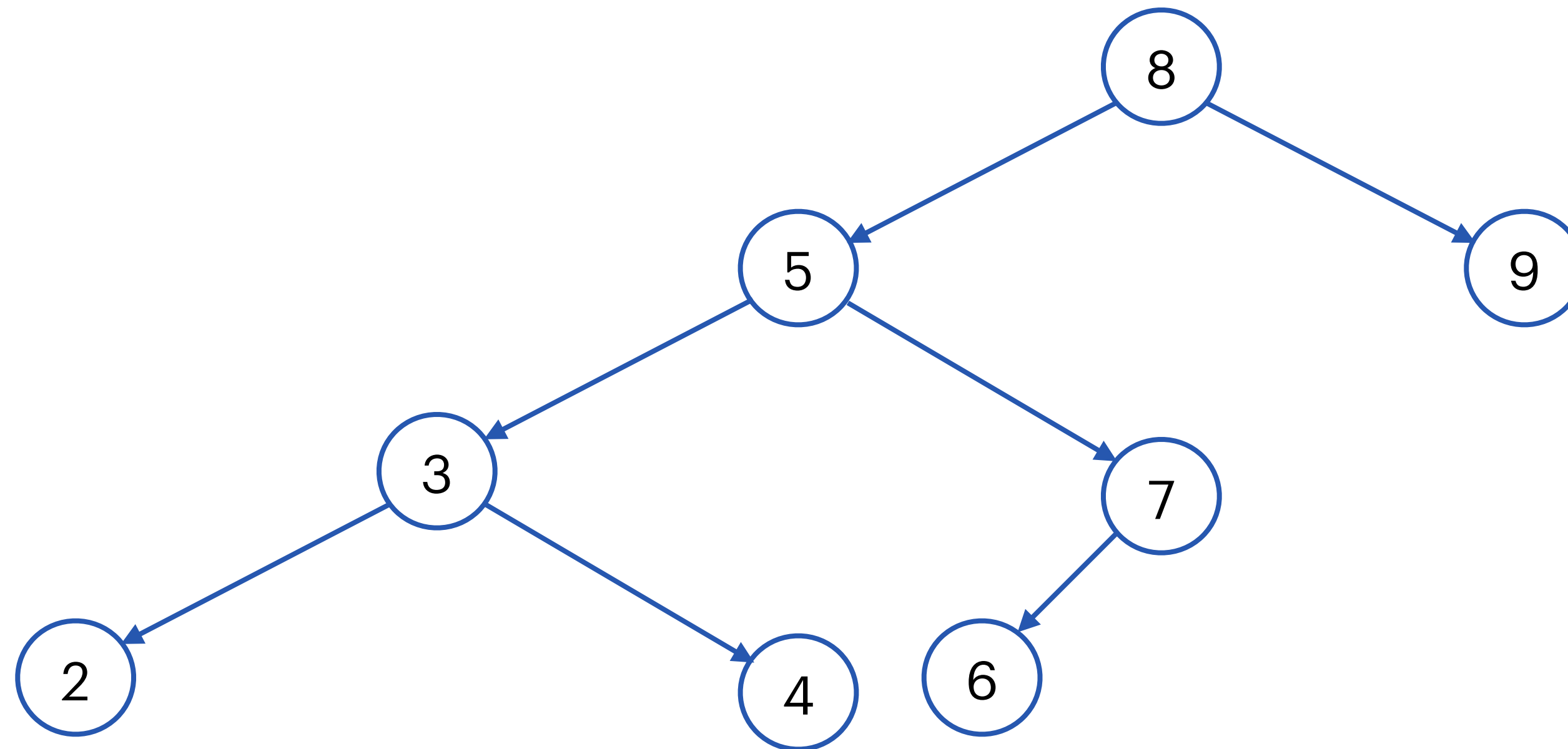
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există



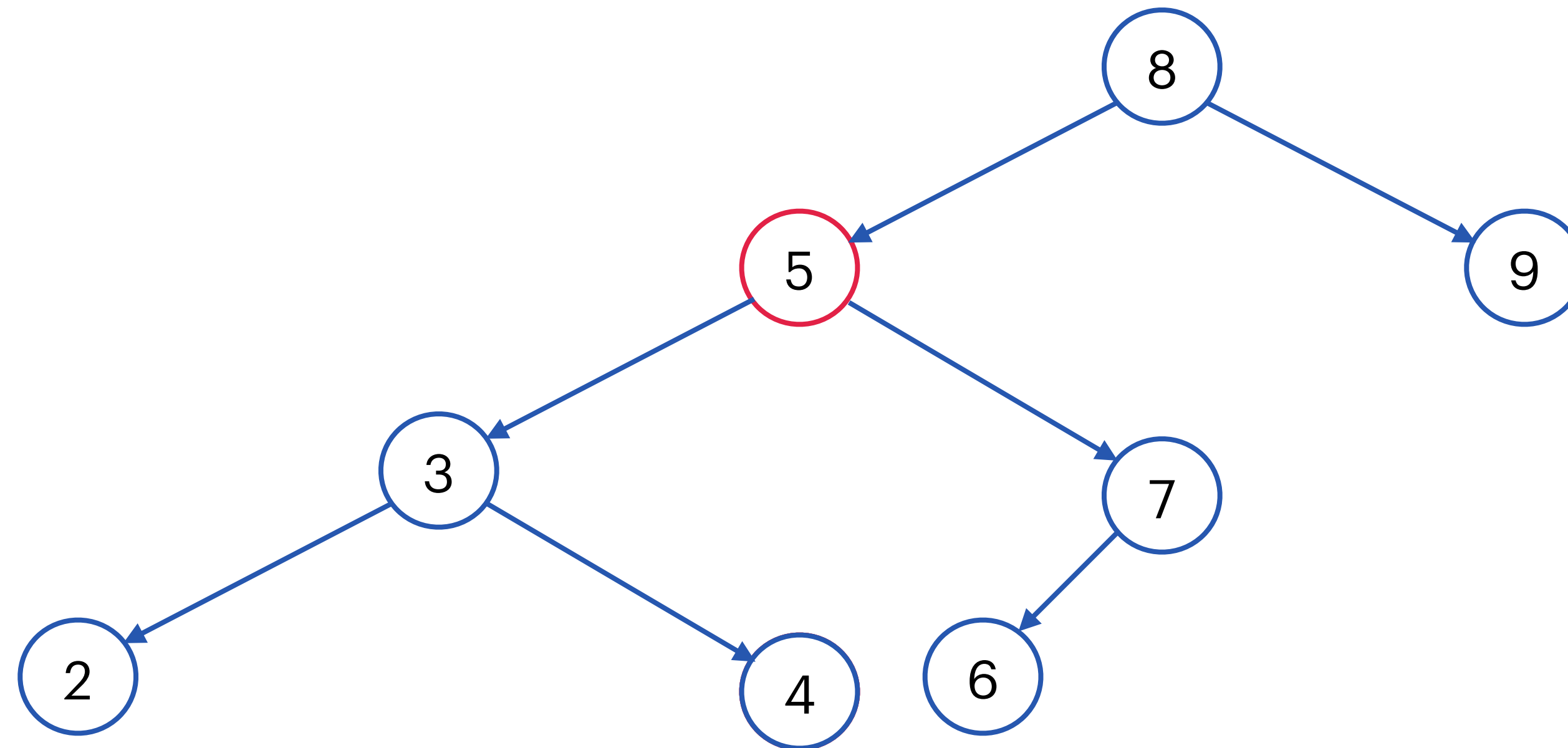
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



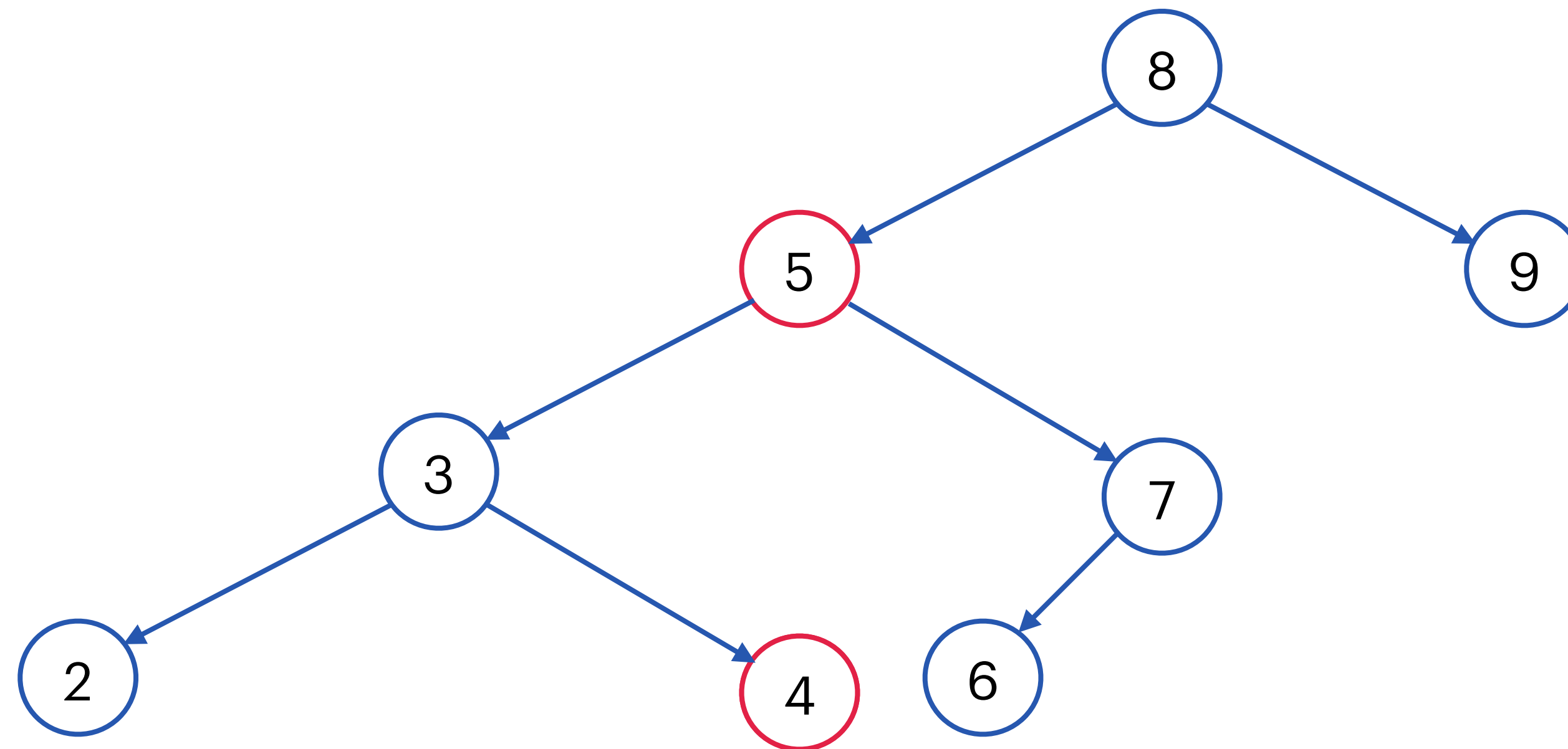
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



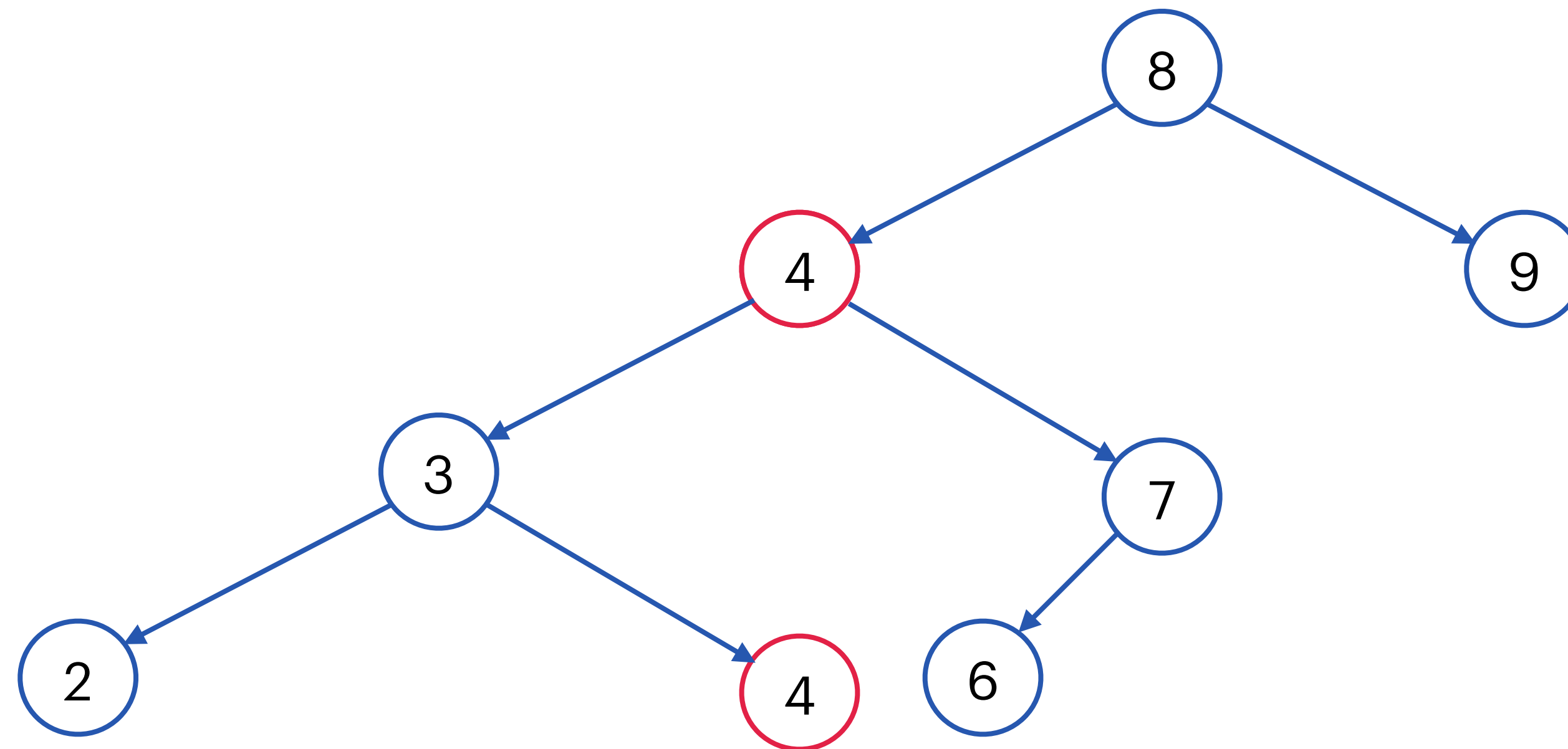
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



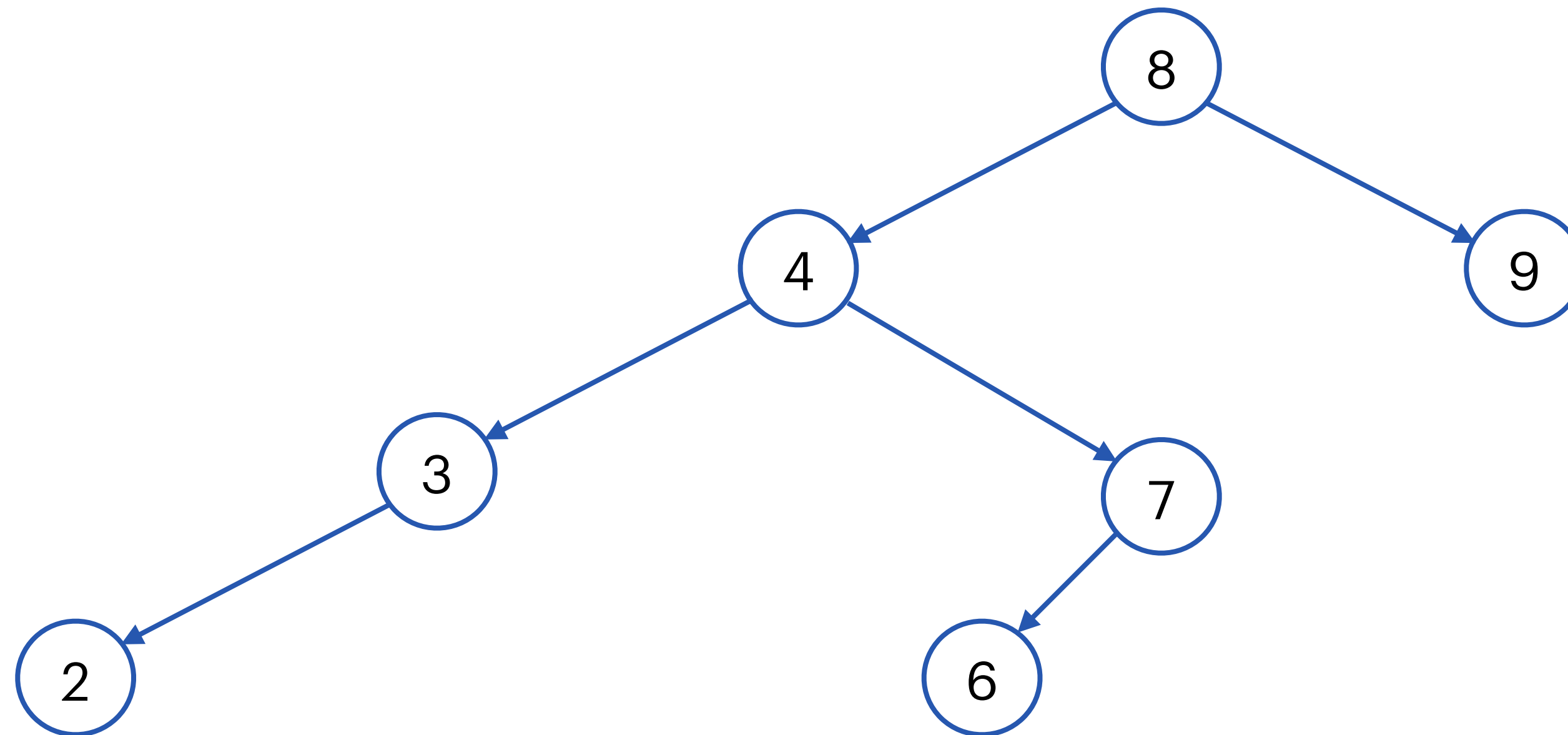
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



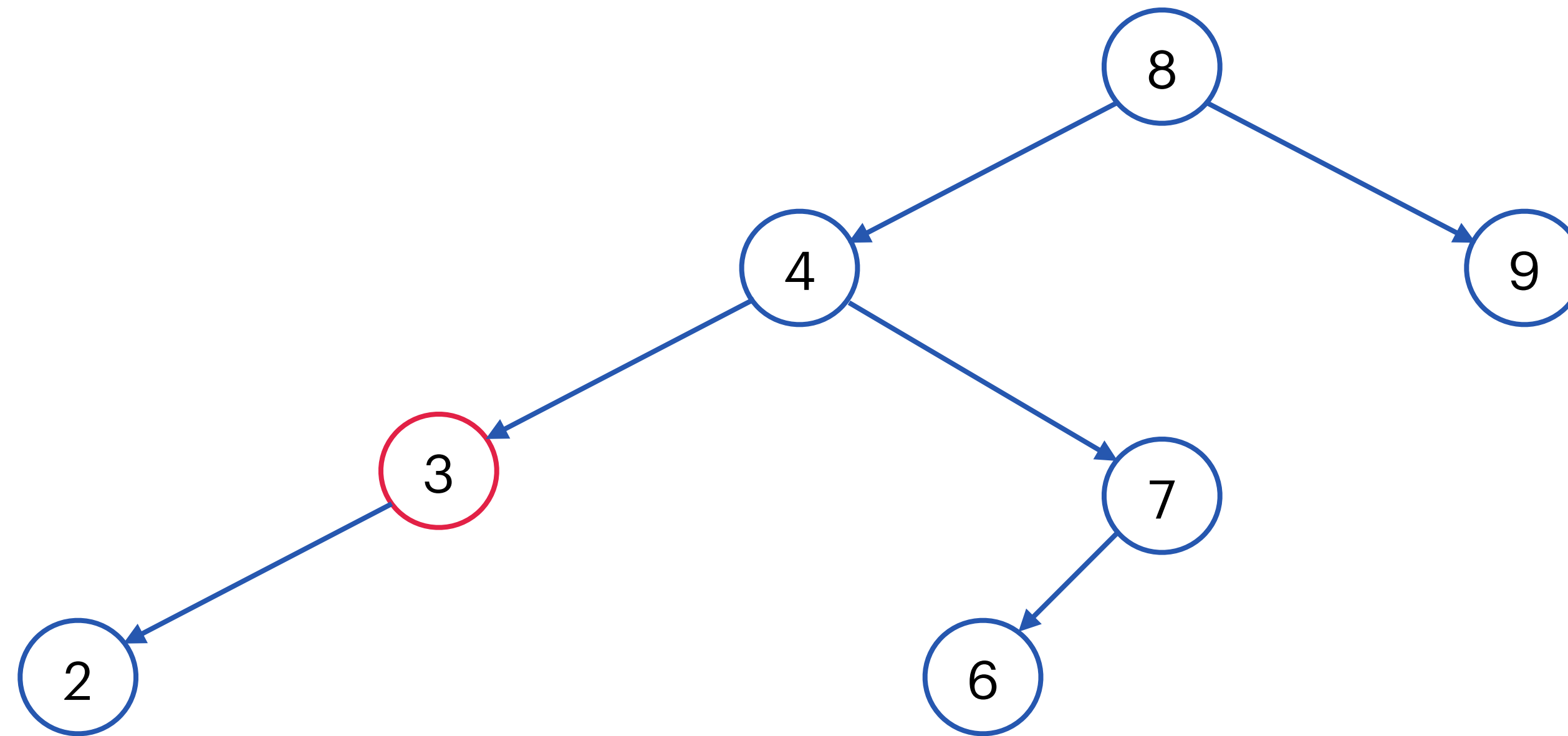
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



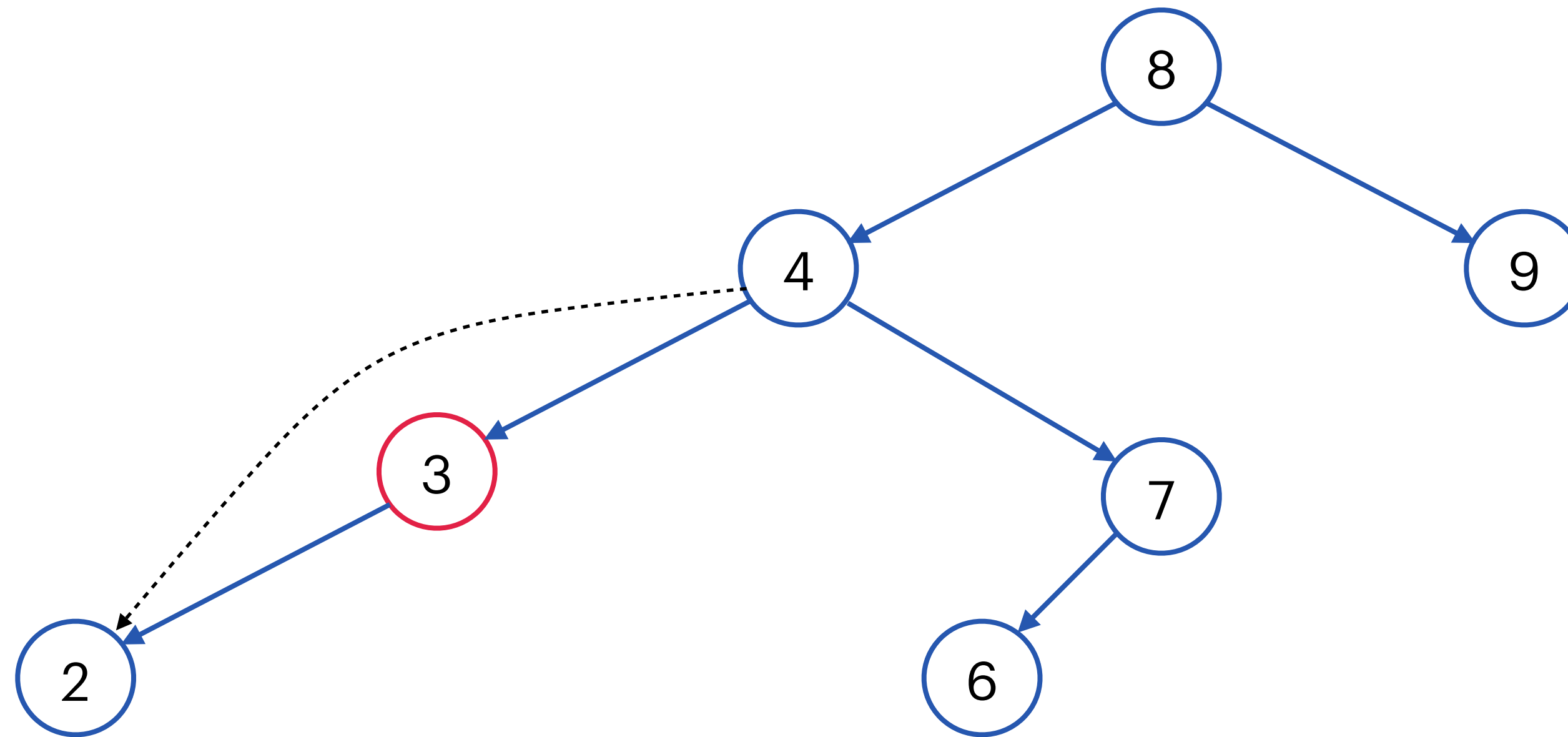
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



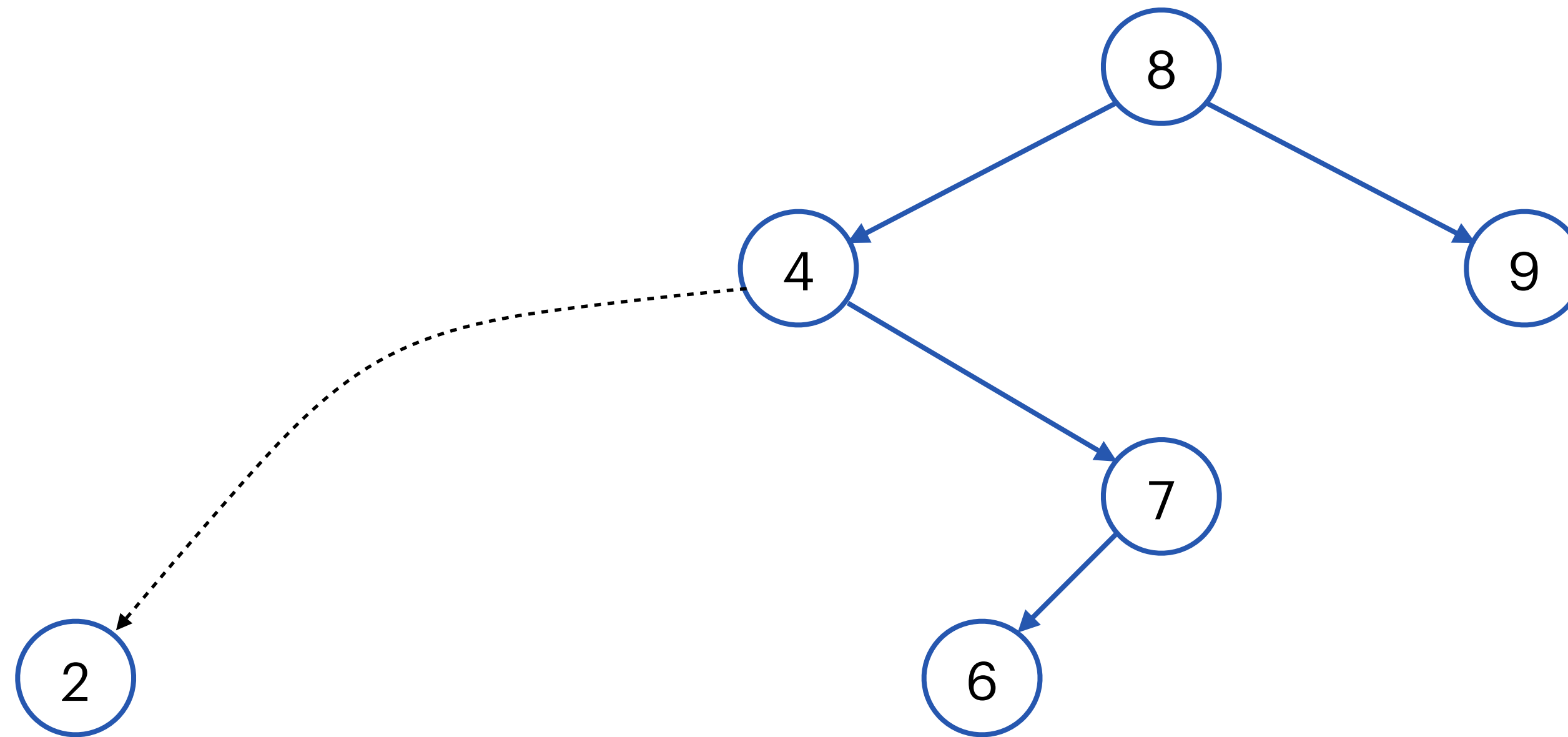
Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.



Arbori Binari Ordonati

Suprimarea nodurilor într-un arbore binar ordonat

În urma suprimării, arborele trebuie să rămână ordonat.

Două cazuri:

1. Nodul de suprimat are cel mult fiu: tatăl nodului de suprimat va indica către unicul fiu dacă acesta există
2. Nodul de suprimat are ambii fii. Se înlocuiește nodul de suprimat cu predecesorul său în inordine. Se suprimă predecesorul care prin definiție nu are fiu drept.

