

# Синхронизиране на успоредно изпълнение

# Преглед

- ▶ Успоредно изпълнение на транзакции
- ▶ Проблеми
  - ▶ Dirty read, unrepeatable read, phantom
  - ▶ Решения - заключване на ресурси
- ▶ Прецизност на заключването
  - ▶ Управление на конкуренцията в B+ дървета
- ▶ Още проблеми - застой
  - ▶ Решения
- ▶ Алтернативи на заключването на ресурси

# Успоредно изпълнение на транзакции

- ▶ Защо? - За да се повиши продуктивността на базата.

$T1$	$T2$
$R(A)$	
$W(A)$	
	$R(B)$
	$W(B)$
$R(C)$	
$W(C)$	

# Транзакции в SQL

- ▶ Характеристики
- ▶ Access mode
  - ▶ READ ONLY | READ WRITE
- ▶ Isolation level
- ▶ Проблемы

Level	Dirty Read	Unrepeatable Read	Phantom
READ UNCOMMITTED	Maybe	Maybe	Maybe
READ COMMITTED	No	Maybe	Maybe
REPEATABLE READ	No	No	Maybe
SERIALIZABLE	No	No	No

# Dirty read $\neq$ Unrepeatable read

**Transaction A**



write (where  $x = 1$ )

rollback

Record in Transaction B is now dirty

**Transaction B**



read (where  $x = 1$ )

**Transaction A**



read (where  $x = 1$ )

read (where  $x = 1$ )

Transaction A might get a record with different values between reads

**Transaction B**



write (where  $x = 1$ )

commit

# Проблемът фантом

Име	Години	Ранг
Тодор	78	1
Стамат	54	1
Пешо	56	2
Минчо	66	2
Жоро	72	1

# Проблемът фантом

Име	Години	Ранг
Тодор	78	1
Стамат	54	1
Пешо	56	2
Минчо	66	2
Жоро	72	1

# Проблемът фантом

Име	Години	Ранг
Тодор	78	1
Стамат	54	1
Пешо	56	2
Минчо	66	2
Жоро	72	1
Харалампи	88	1



# Проблемът фантом

Име	Години	Ранг
Тодор	78	1
Стамат	54	1
Пешо	56	2
Минчо	66	2
Жоро	72	1
Харалампи	88	1

Резултат = 78

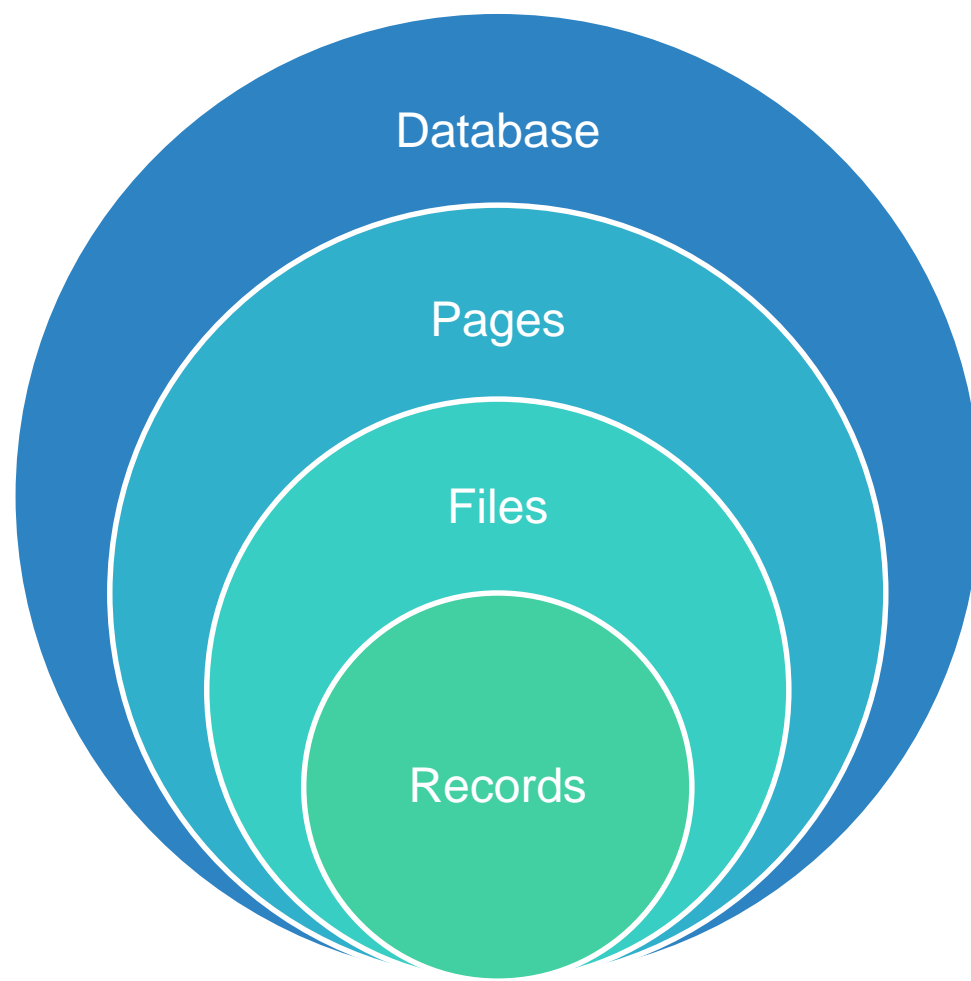
# Заклучване на ресурси

- ▶ Как се осъществява заключването?
- ▶ Процесът се ръководи от Lock manager
  - ▶ Lock table -> lock table entry
  - ▶ Transaction table
- ▶ Атомарност

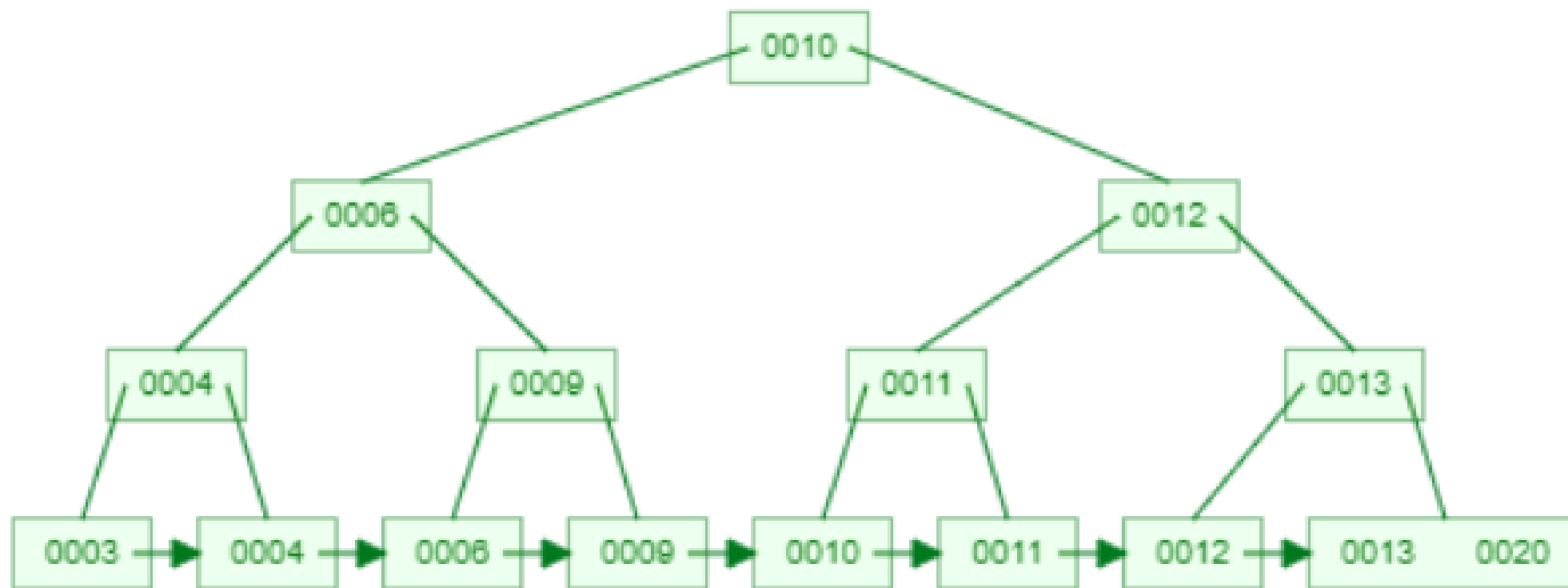
Level	Dirty Read	Unrepeatable Read	Phantom
READ UNCOMMITTED	Maybe	Maybe	Maybe
READ COMMITTED	No	Maybe	Maybe
REPEATABLE READ	No	No	Maybe
SERIALIZABLE	No	No	No

# Прецизност на заключването

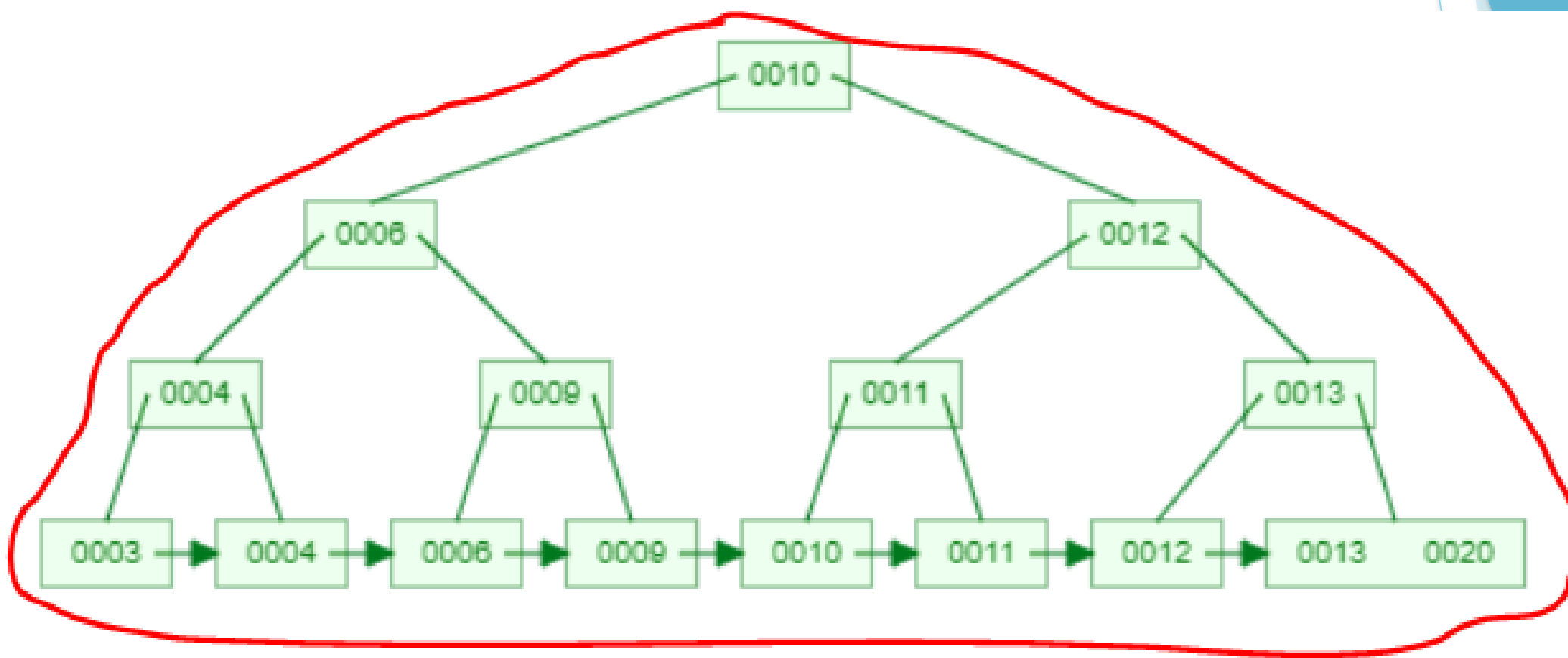
- ▶ Ниво/детайлност на заключването
- ▶ Използване на йерархията на базата



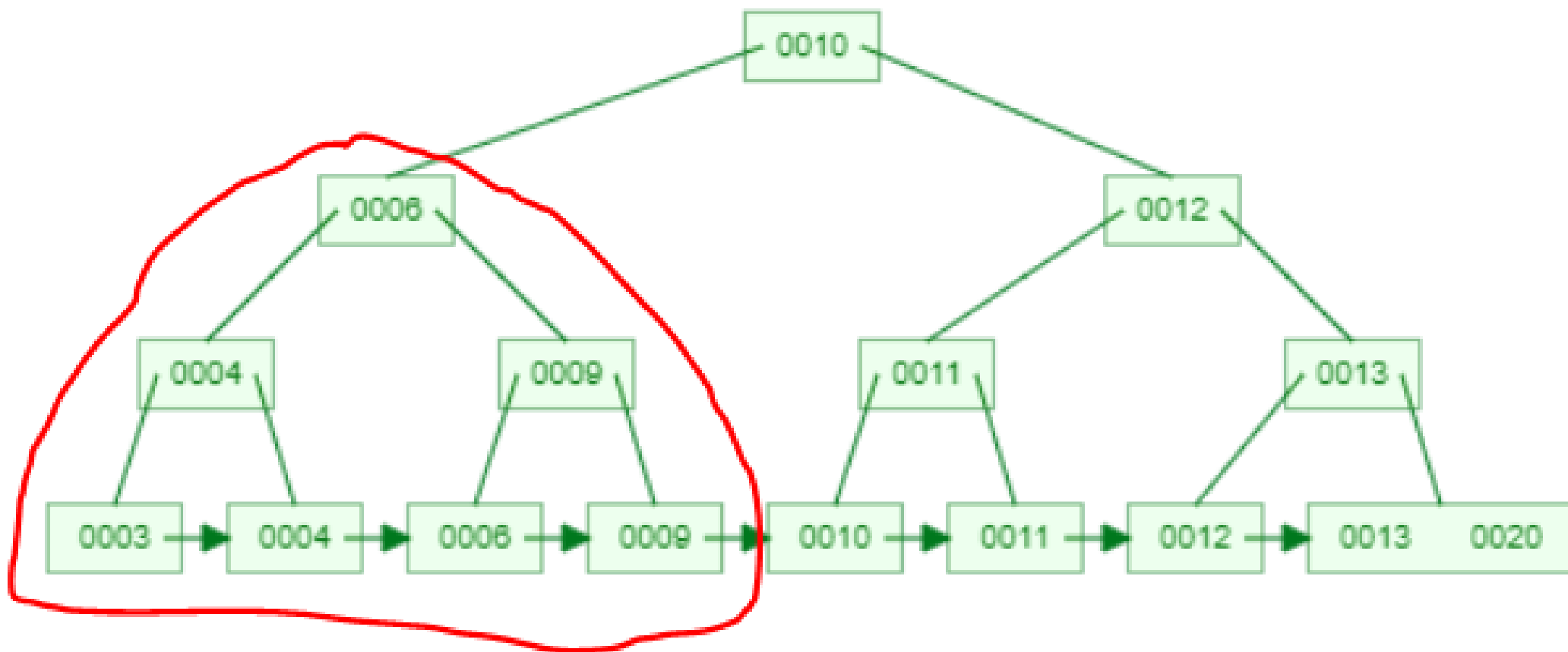
# Управление на конкуренцията в B+ дървета



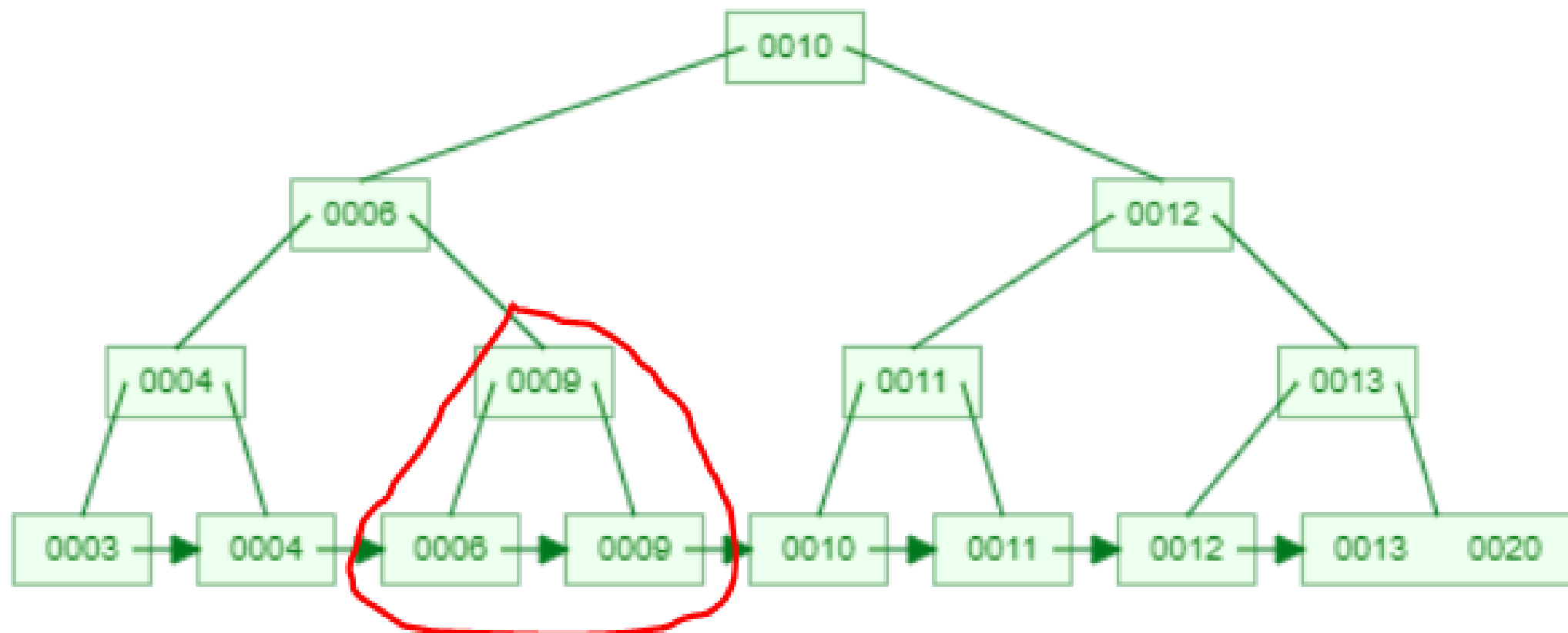
## Управление на конкуренцията в B+ дървета



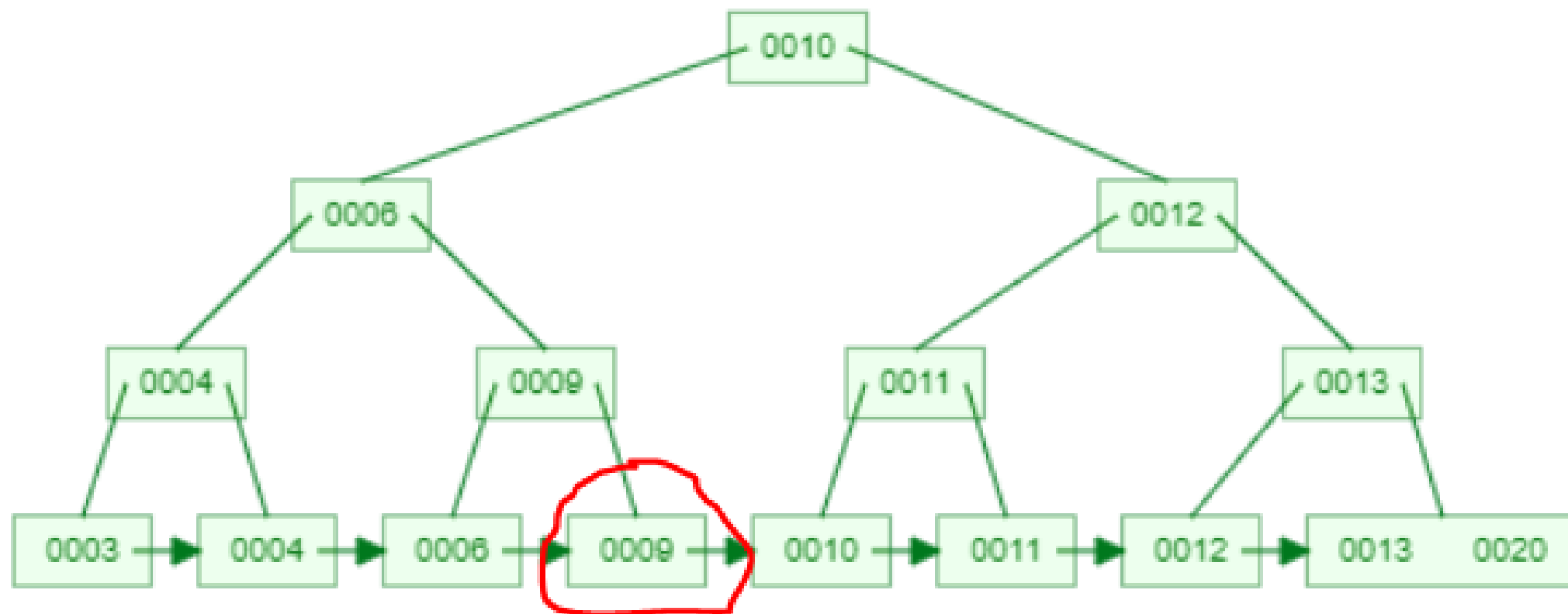
## Управление на конкуренцията в B+ дървета



## Управление на конкуренцията в B+ дървета

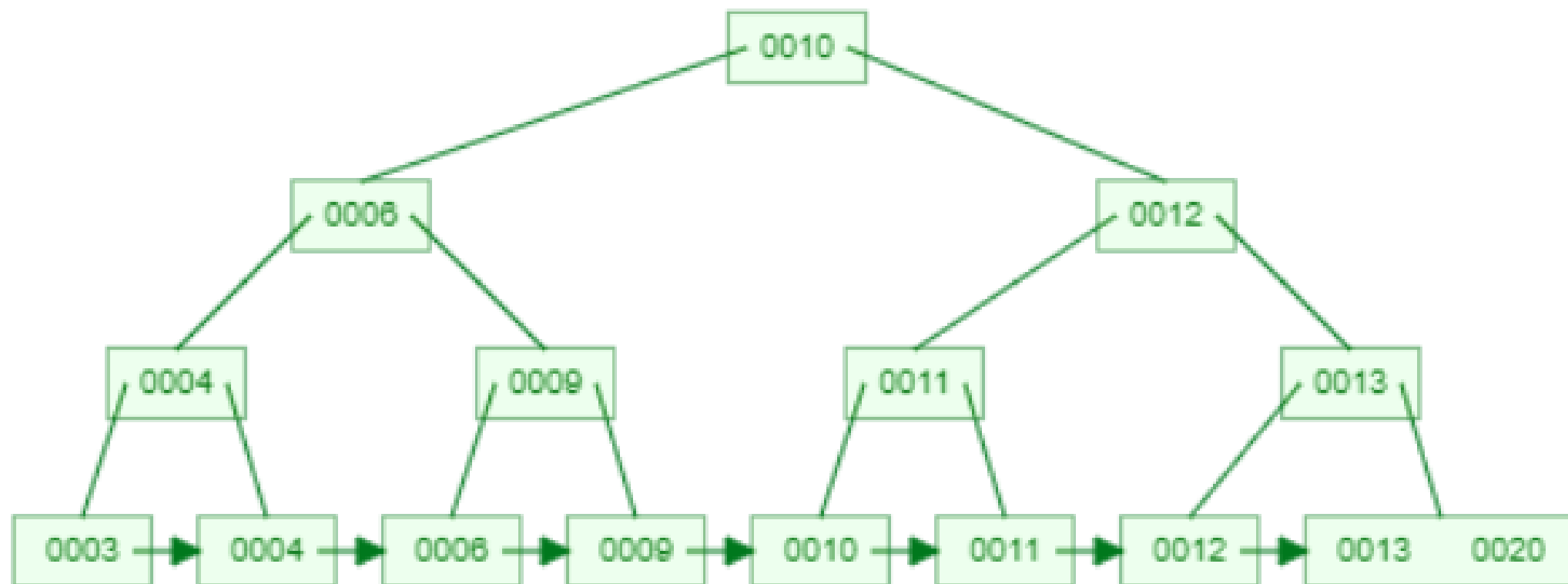


## Управление на конкуренцията в B+ дървета

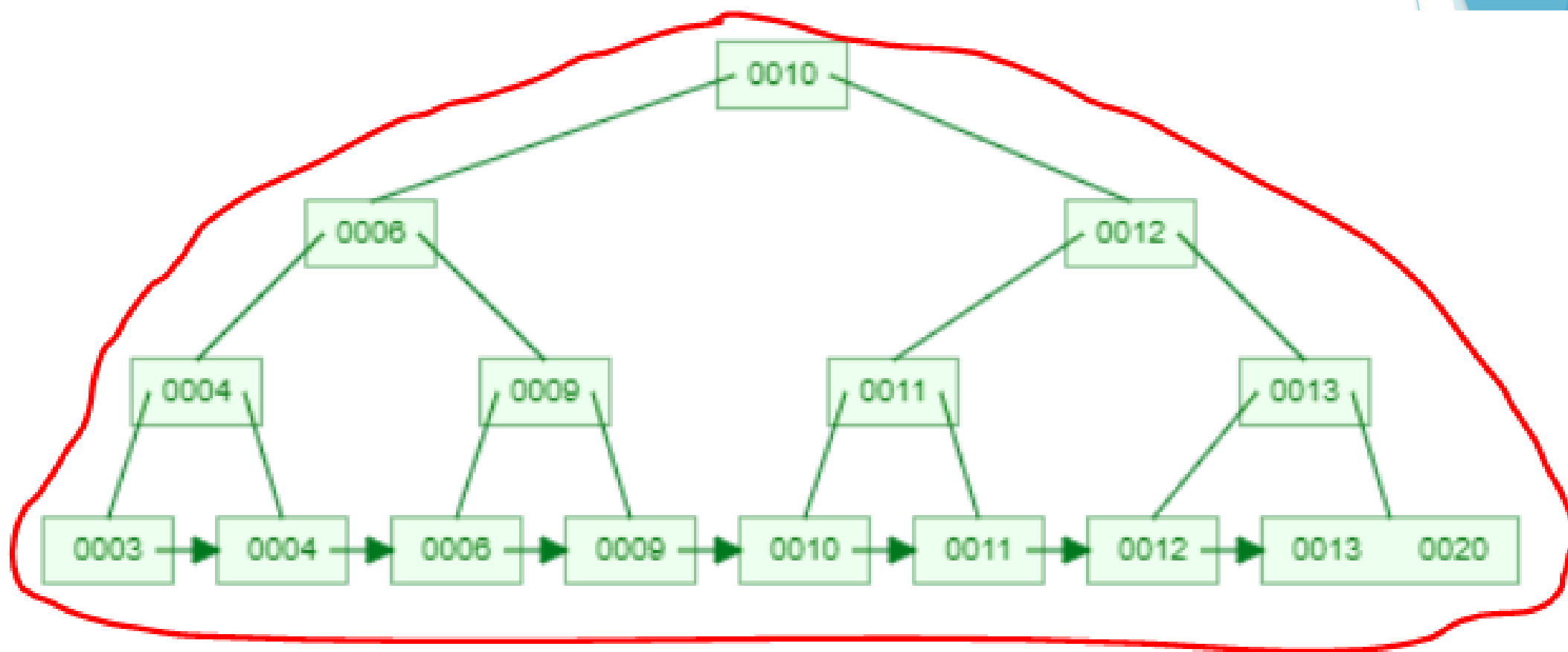




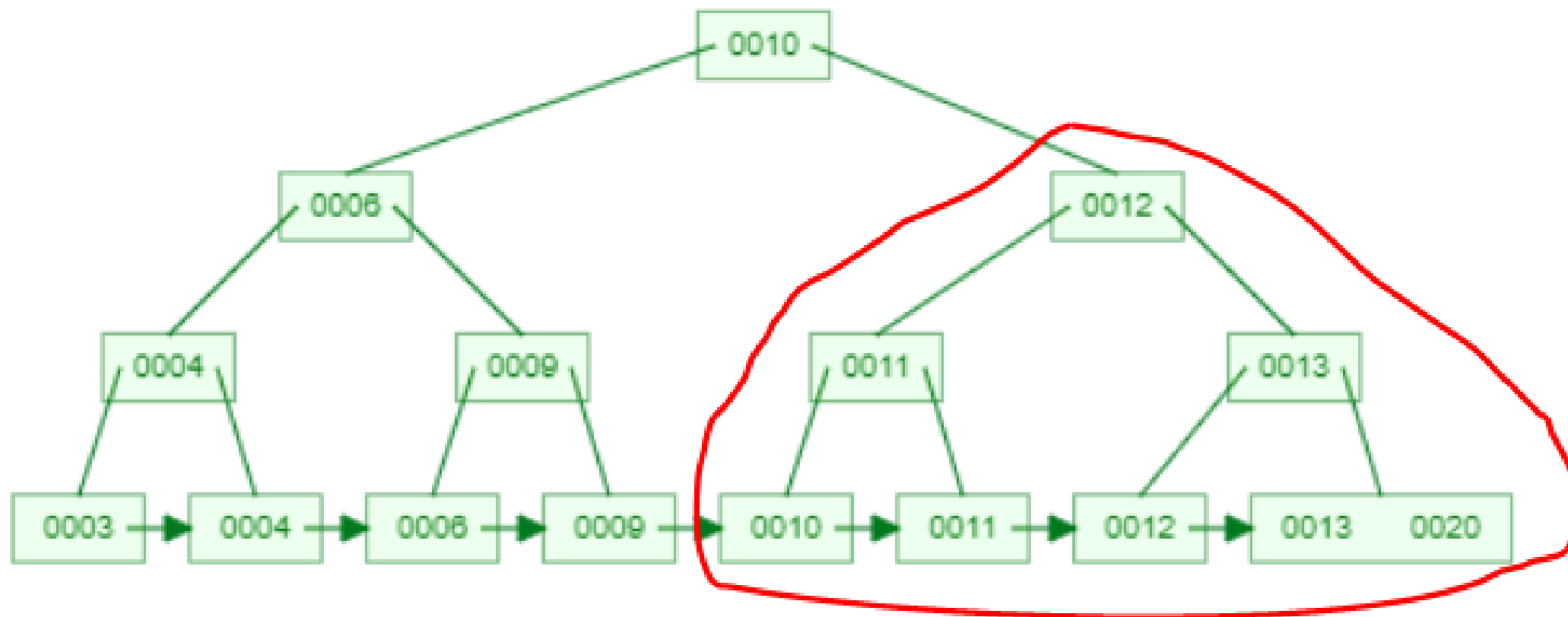
# Управление на конкуренцията в B+ дървета



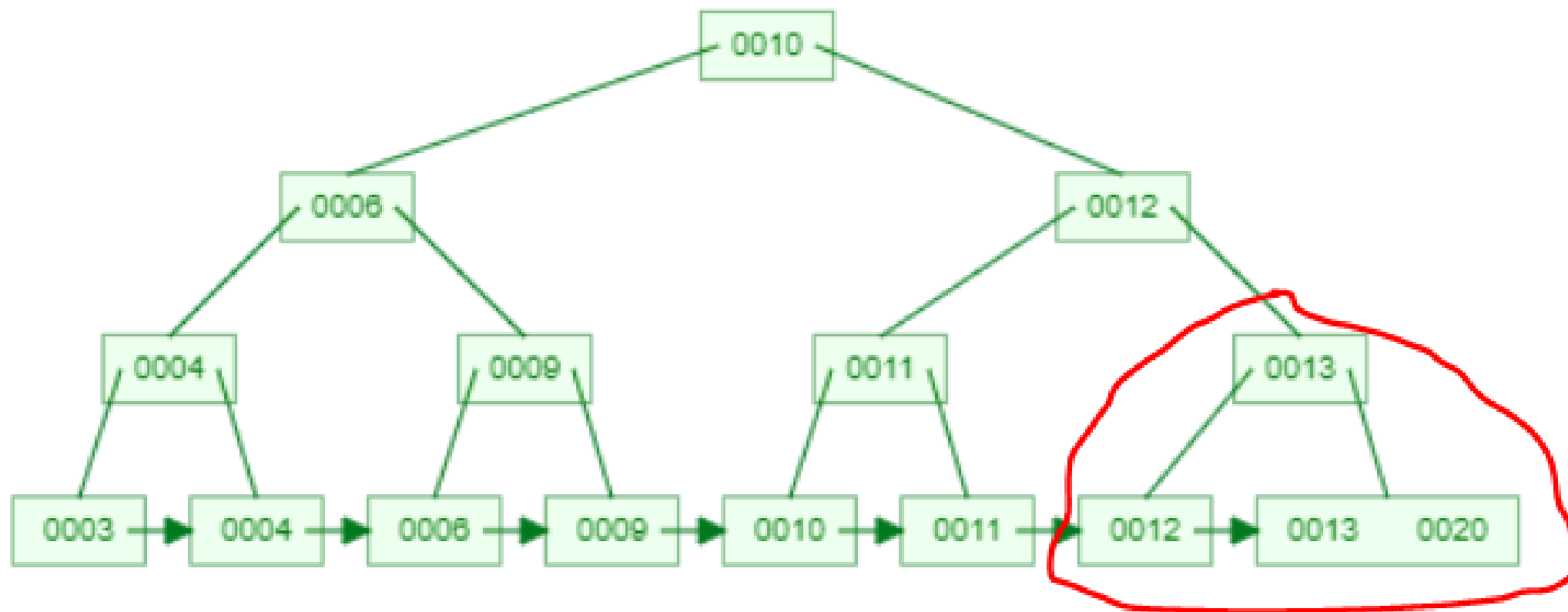
## Управление на конкуренцията в B+ дървета



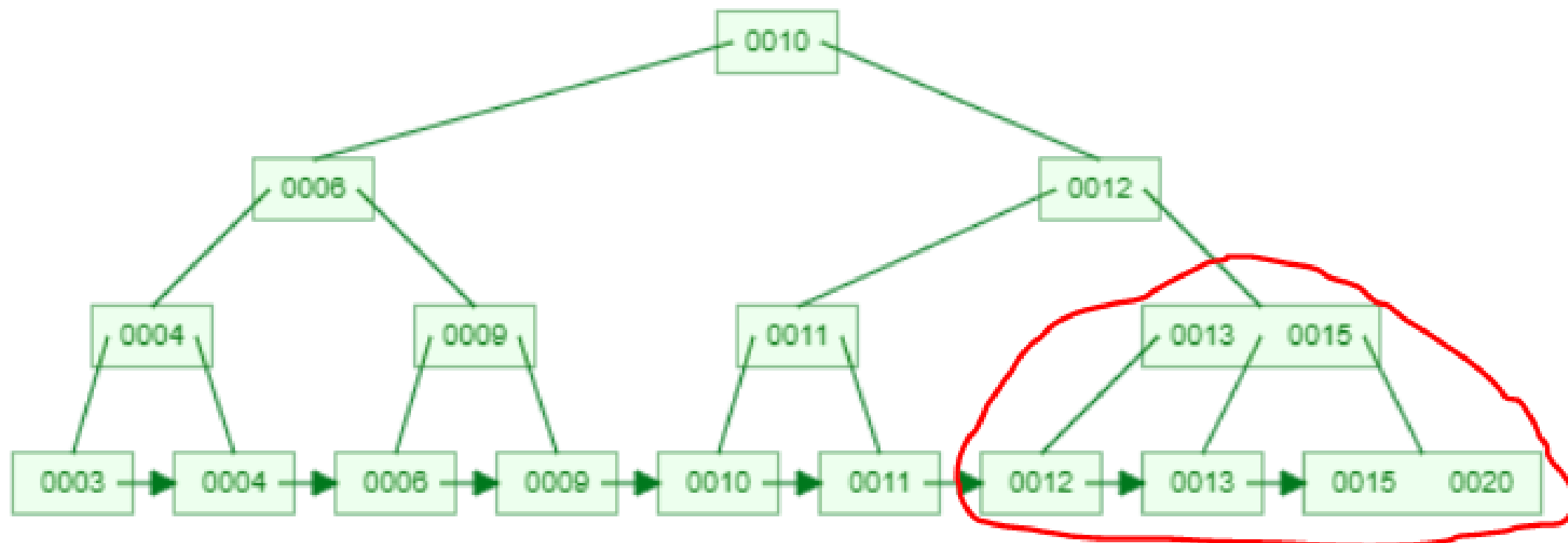
# Управление на конкуренцията в B+ дървета



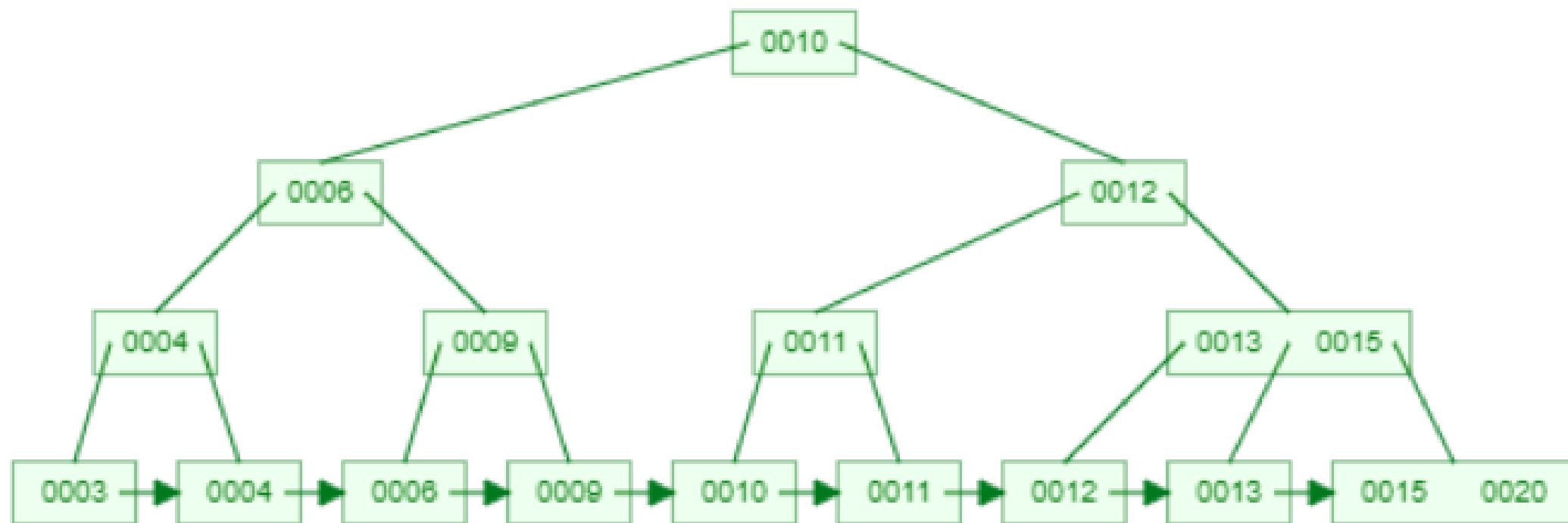
# Управление на конкуренцията в B+ дървета



# Управление на конкуренцията в B+ дървета

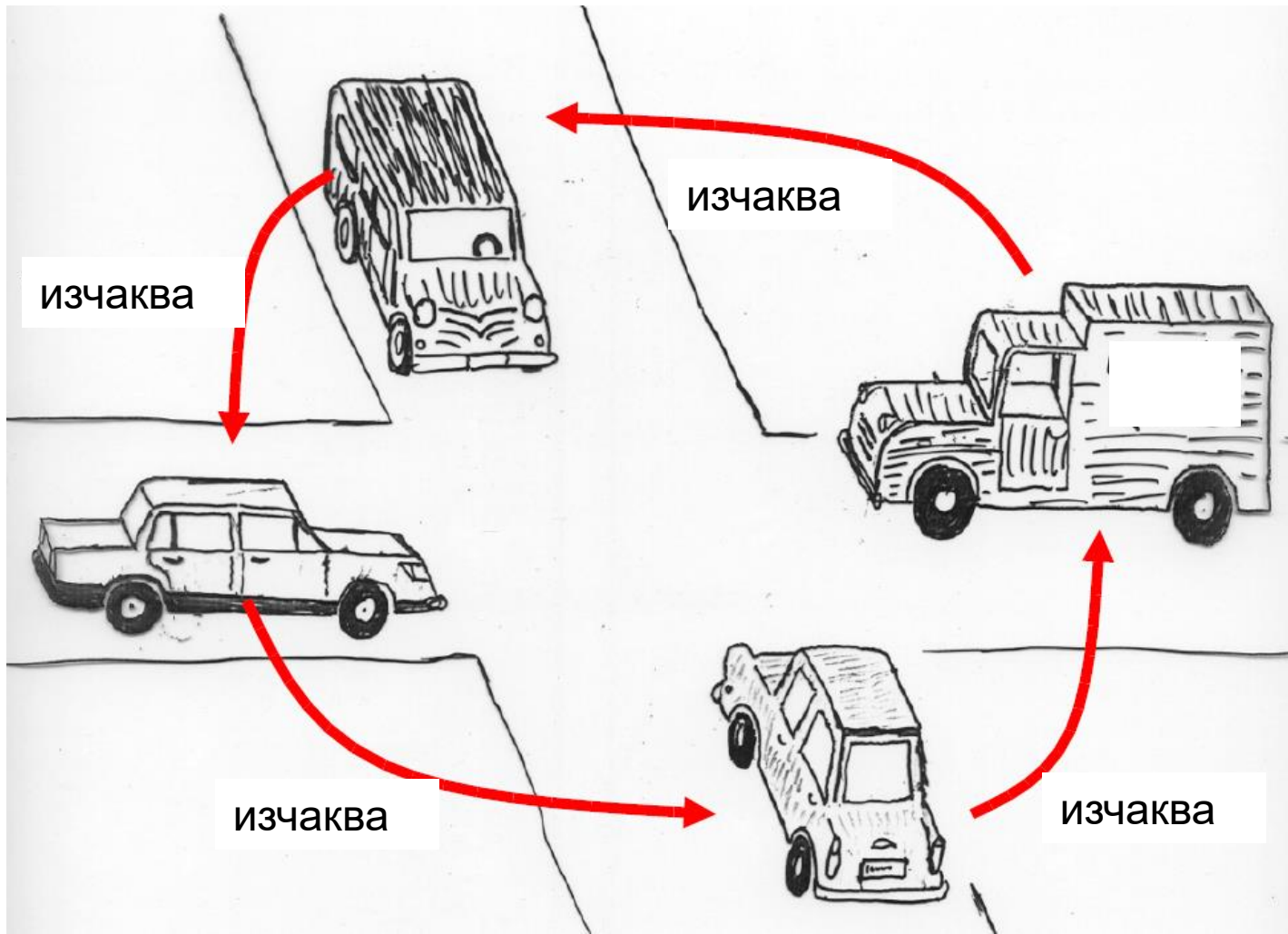


# Управление на конкуренцията в B+ дървета



# Застой

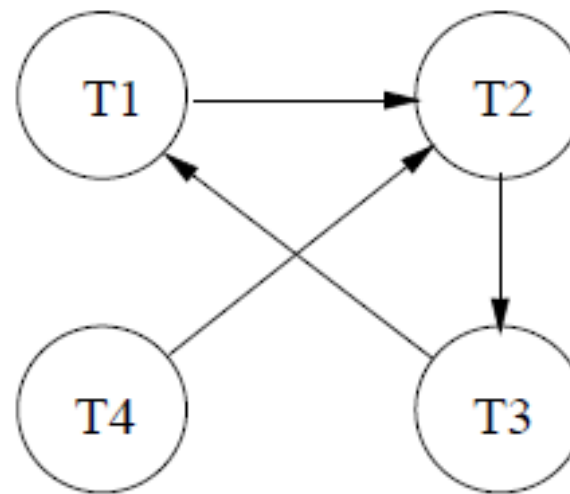
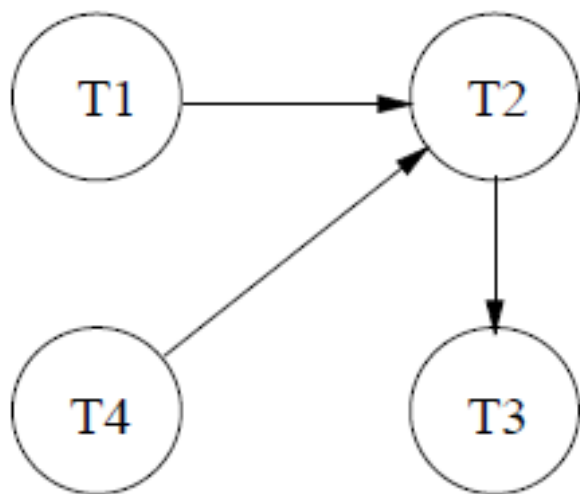
- ▶ Какво е застой в сферата на бази данни?







- ▶ Избягване
  - ▶ Чрез даване на приоритет на всяка транзакция спрямо времето на пристигане
- ▶ Разпознаване
  - ▶ Waits-for graphs
  - ▶ Timeout механизъм
- ▶ Избягване срещу разпознаване



# Алтернативи на заключването - никакво заключване

- ▶ Оптимистичен подход
  - ▶ Четене, валидация и писане
- ▶ Timestamp
- ▶ Thomas write rule

$T1$	$T2$
$R(A)$	$W(A)$ Commit
$W(A)$ Commit	

$T1$	$T2$
$R(A)$	Commit
$W(A)$ Commit	