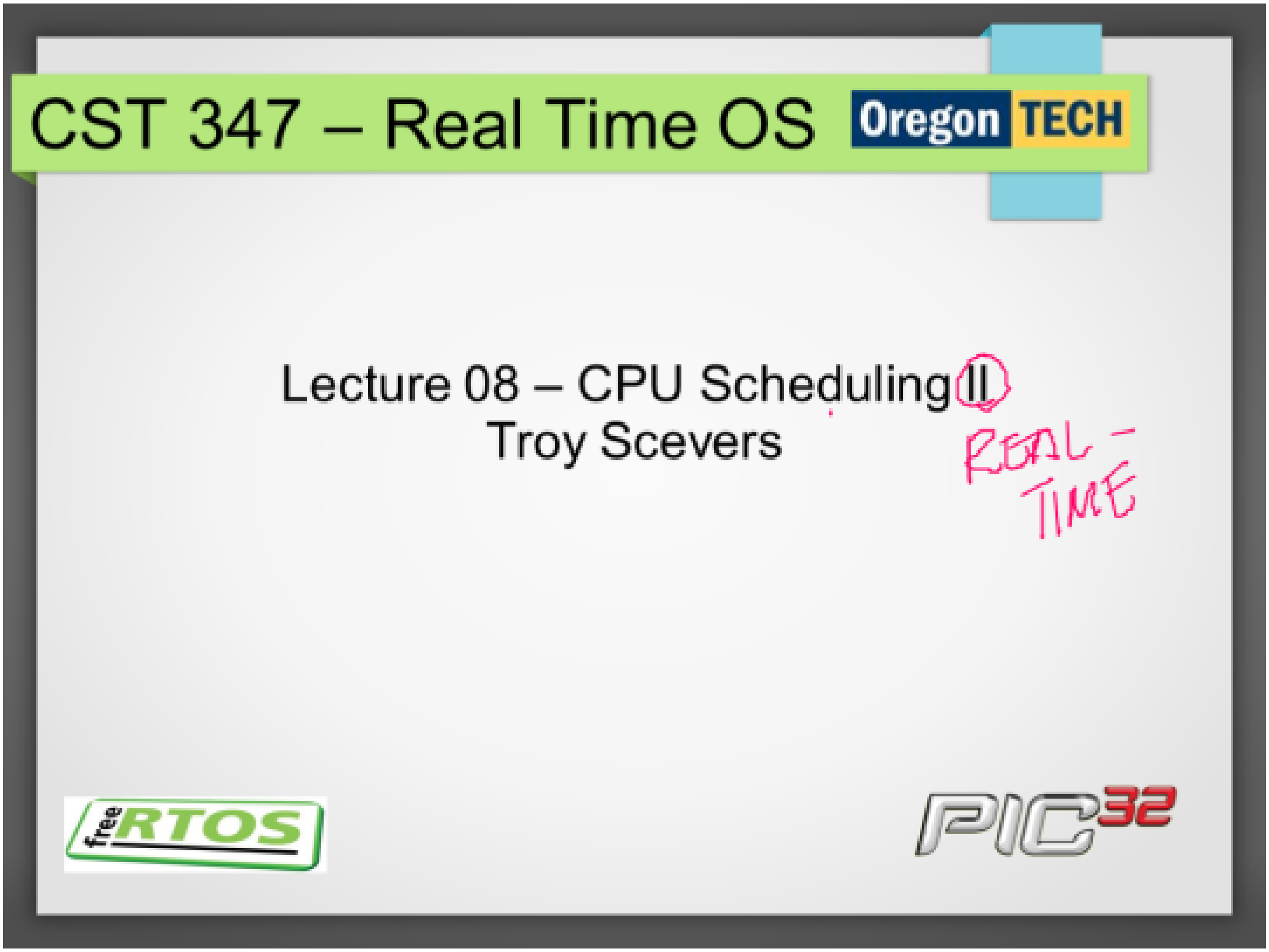
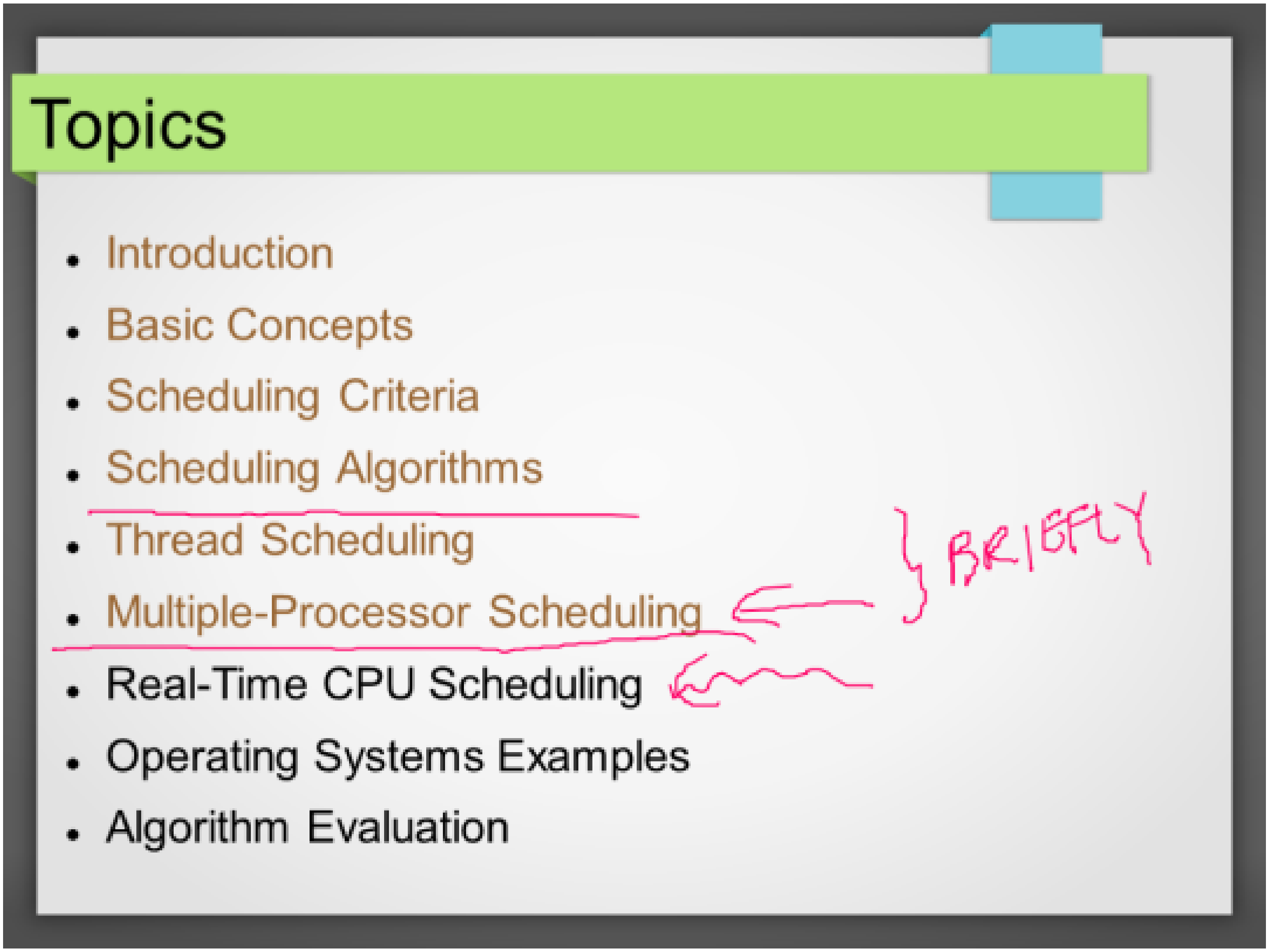
Slide1


CST 347 – Real Time OS
Oregon TECH
Lecture 08 – CPU Scheduling II
Troy Scevers
REAL-TIME
freeRTOS
PIC32

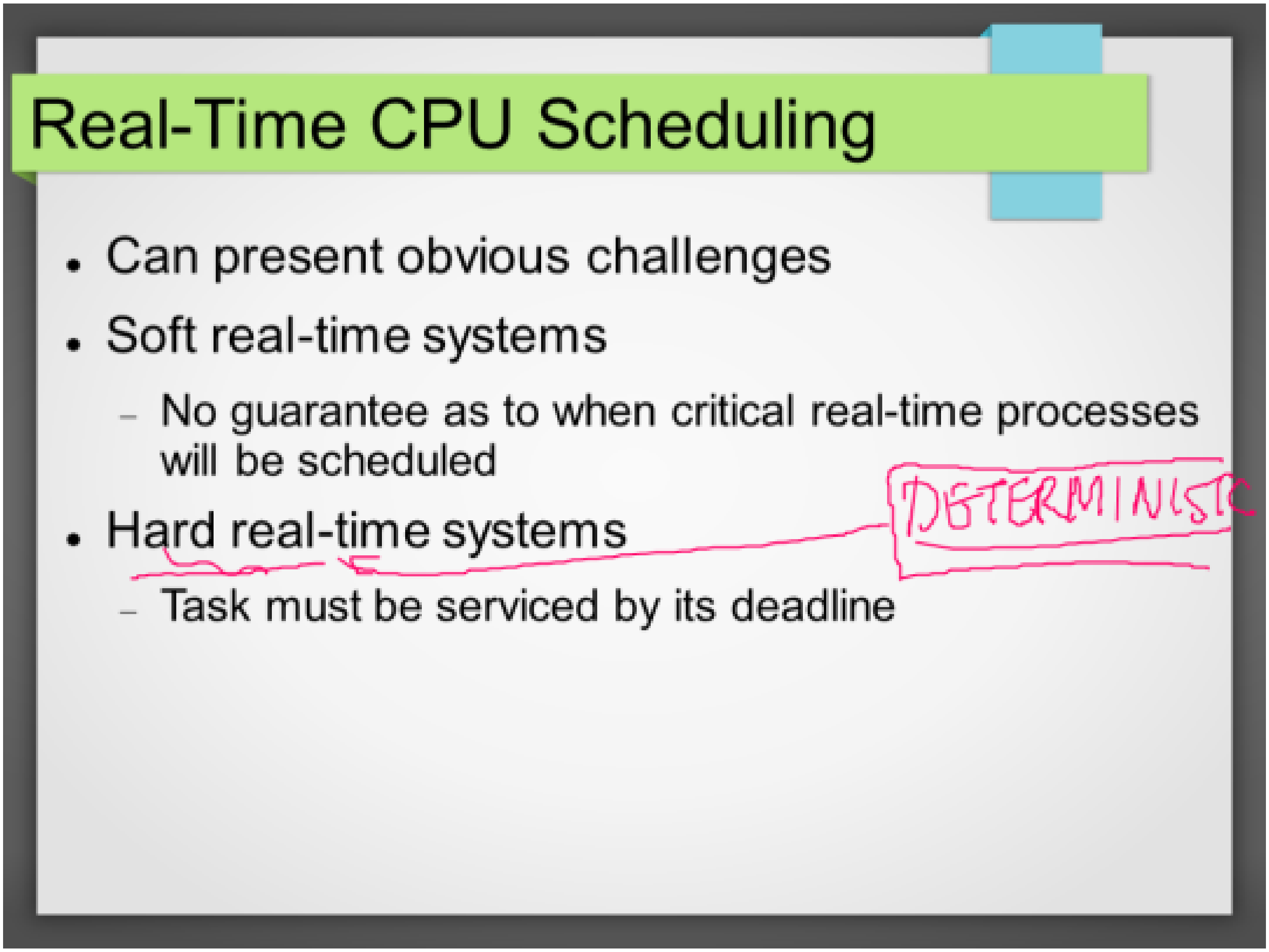
Slide2

CST 347 – Real-Time OS

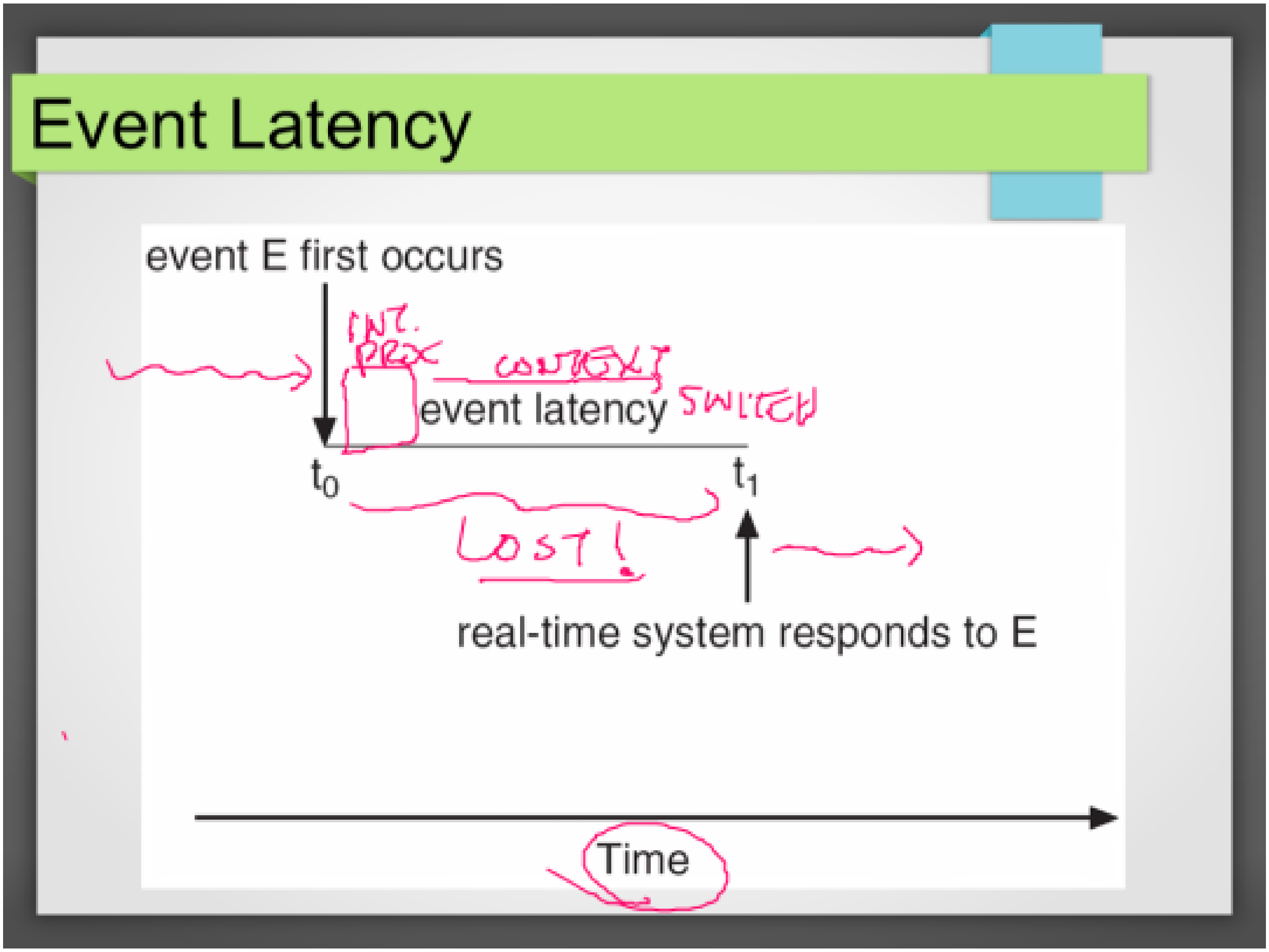
Slide3


Topics
Introduction
Basic Concepts
Scheduling Criteria
Scheduling Algorithms
Thread Scheduling
Multiple-Processor Scheduling
Real-Time CPU Scheduling
Operating Systems Examples
Algorithm Evaluation
BRIEFLY

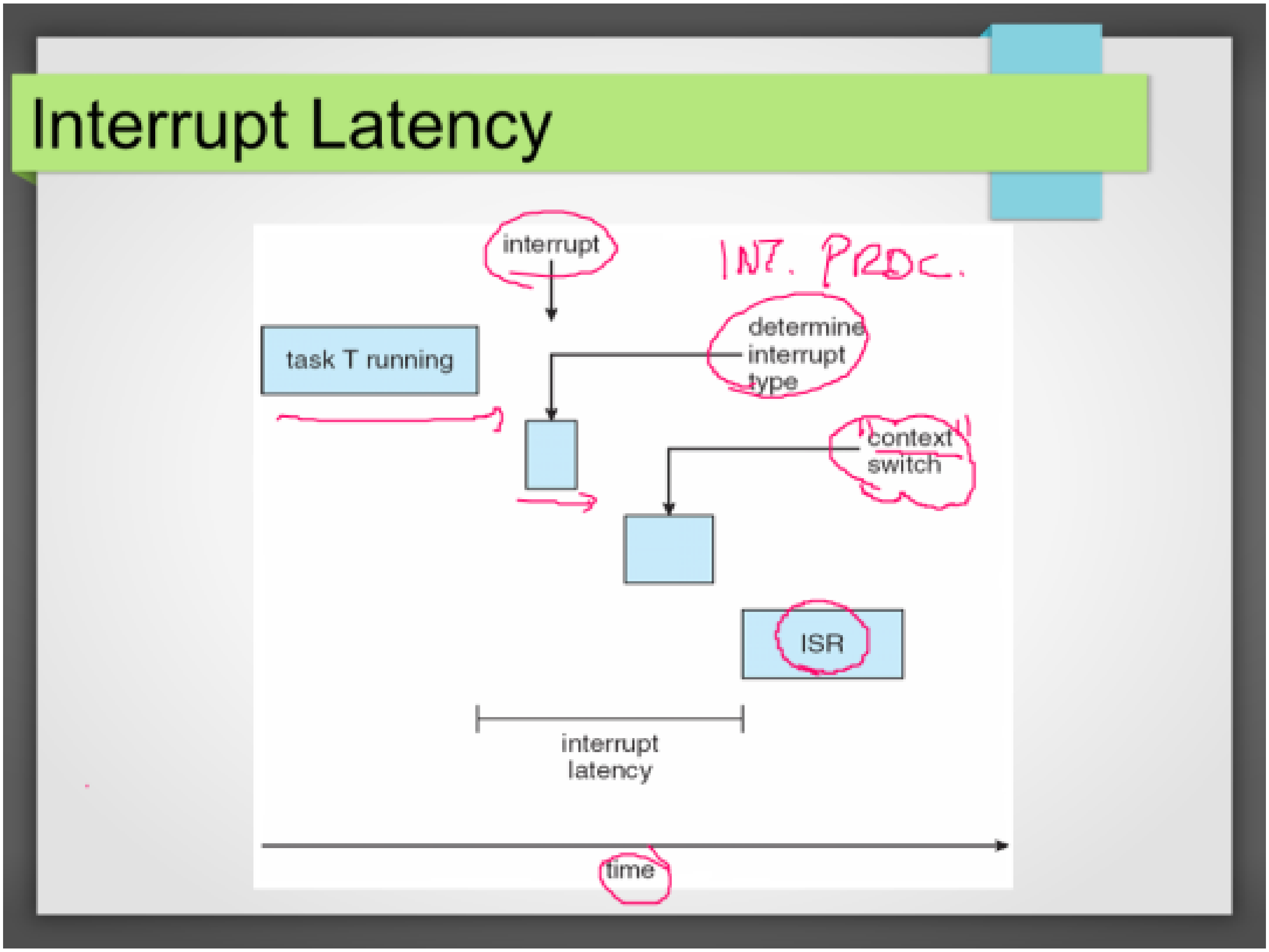
Slide4


Real-Time CPU Scheduling
• Can present obvious challenges
• Soft real-time systems
– No guarantee as to when critical real-time processes will be scheduled
• Hard real-time systems
DETERMINISTIC
– Task must be serviced by its deadline

Slide5


Event Latency
event E first occurs
event latency
t0
t1
real-time system responds to E
Time
CONTEXT SWITCH
LOST!

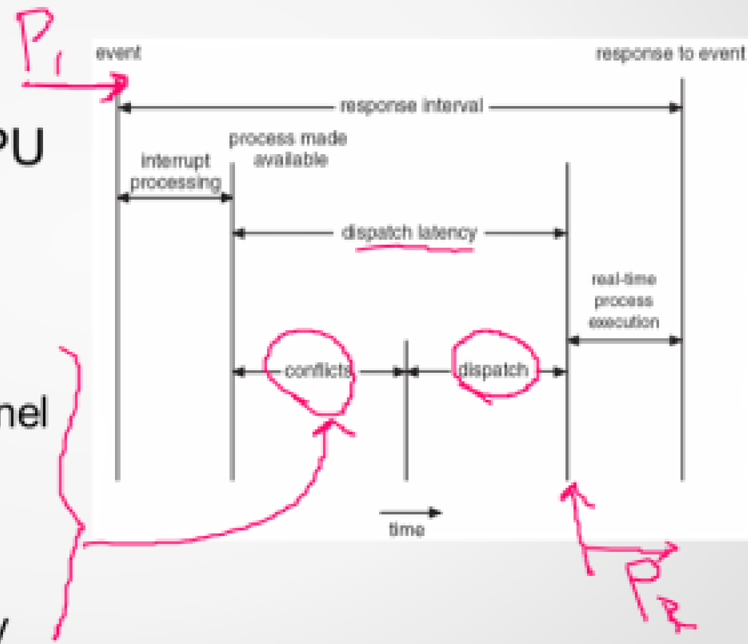
Slide6


Interrupt Latency
interrupt
INT. PROC.
task T running
determine interrupt type
context switch
ISR
interrupt latency
time

Slide7

Dispatch Latency

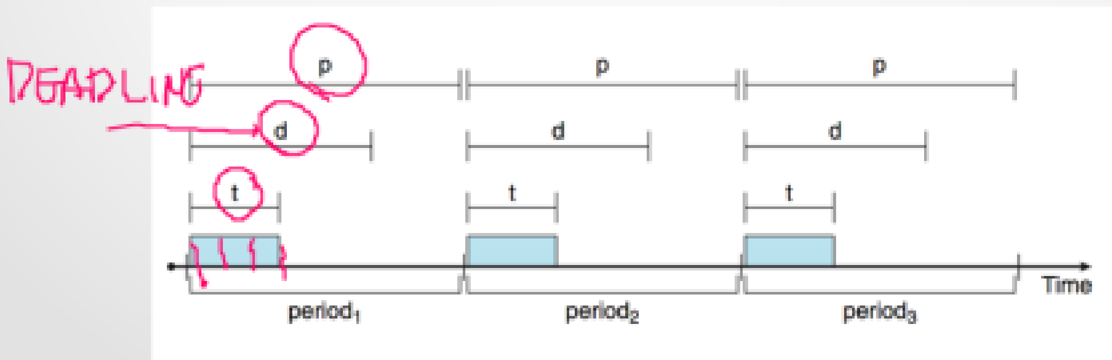
- Time it takes the scheduler to take current process off CPU and switch to another
- Conflict Phase
  - Preemption of any process running in kernel mode
  - Release of low-priority process resources needed by high-priority process



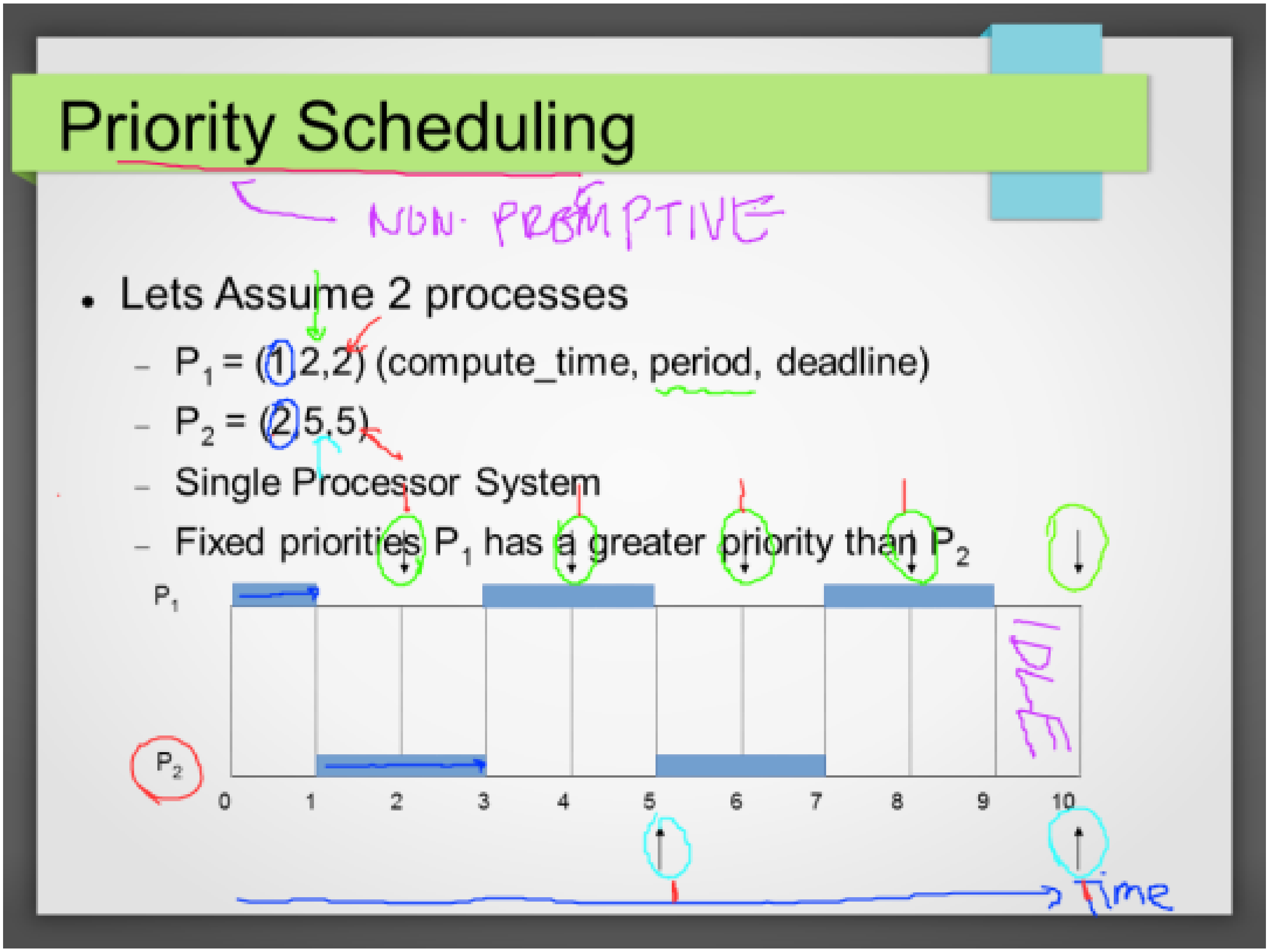
Slide8

Priority Based Scheduling

- For real-time scheduling, scheduler must support preemptive, priority-based scheduling
  - This only guarantee a soft real-time
- Hard real-time must also provide ability to meet deadlines
- Processes have a new characteristic: periodic ones require CPU at constant intervals
  - Has processing time *t*, deadline *d*, period *p*
  - $0 \le t \le d \le p$
  - Rate of periodic tasks is $1/p$



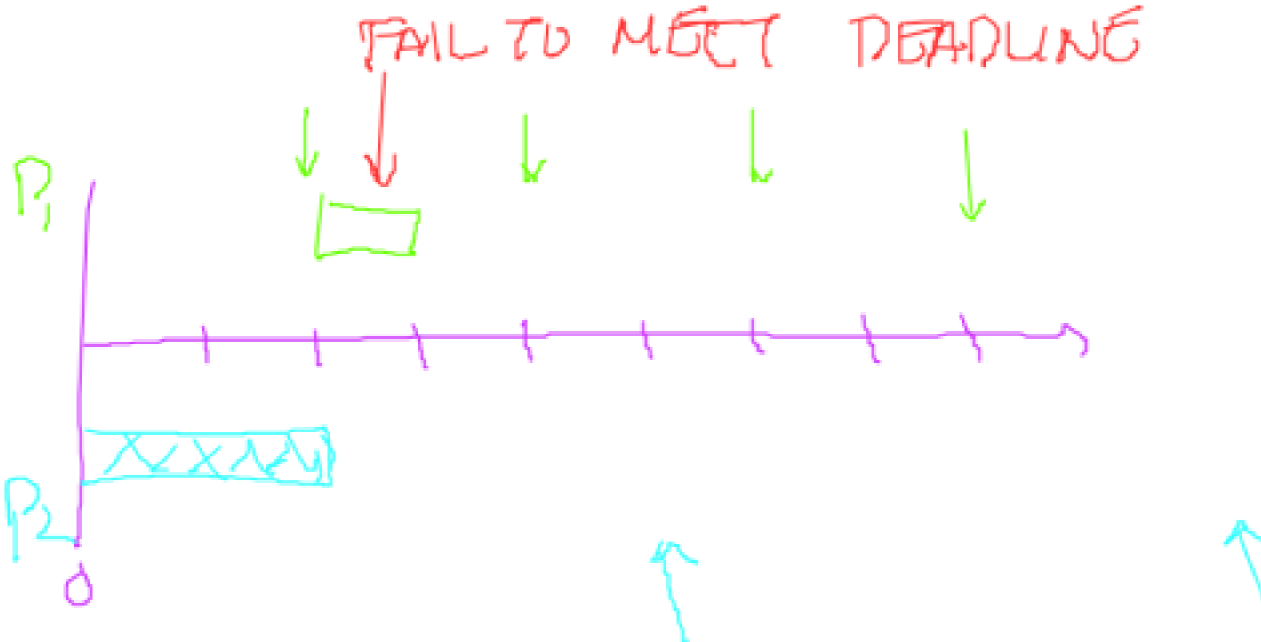
Slide9


Priority Scheduling
NON-PREEMPTIVE
Lets Assume 2 processes
P1 = (1,2,2) (compute_time, period, deadline)
P2 = (2,5,5)
Single Processor System
Fixed priorities P1 has a greater priority than P2
P1
P2
IDLE
0 1 2 3 4 5 6 7 8 9 10
Time

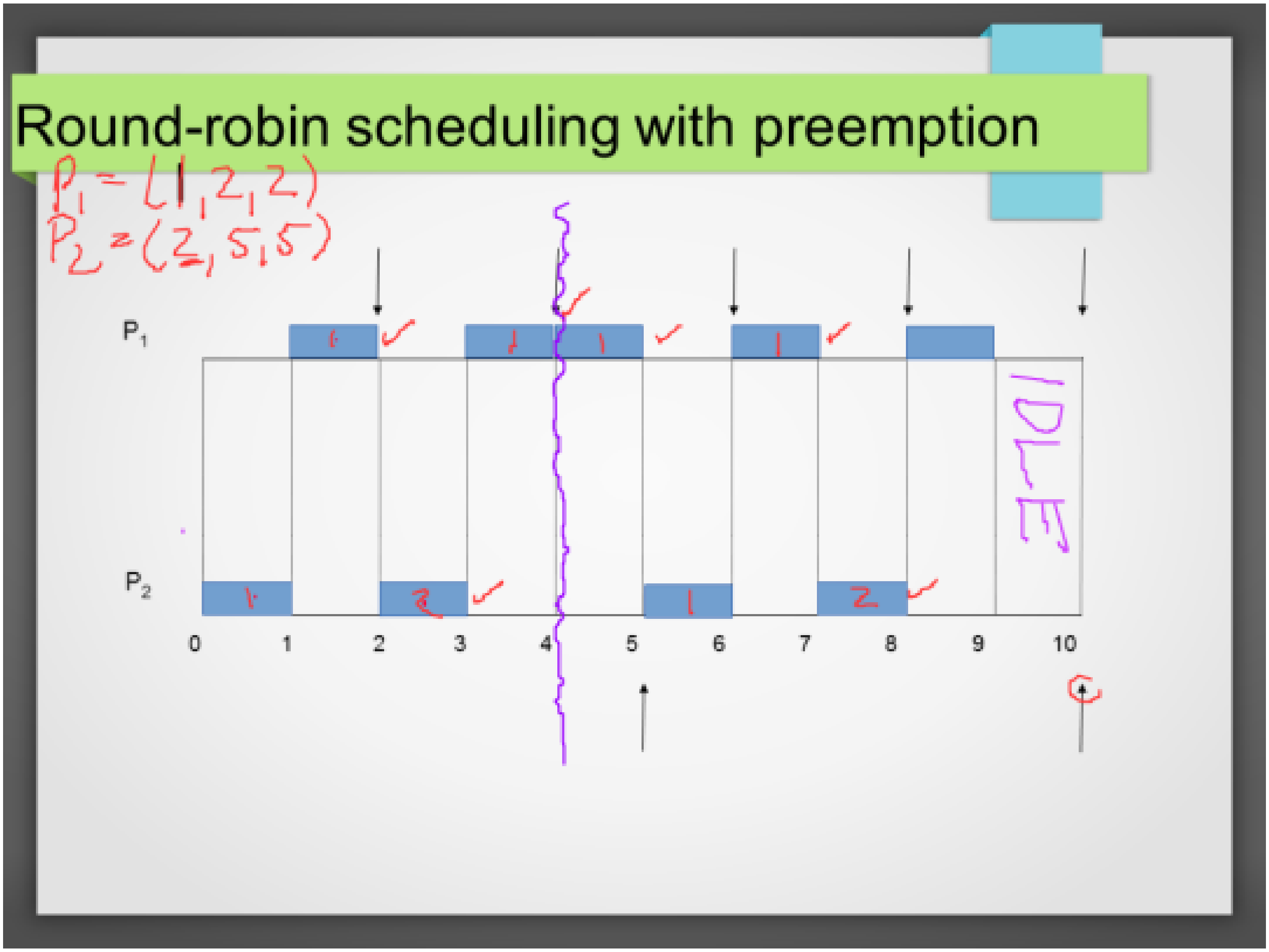
Public Page 1

$P_1 = (1, 2, 2)$

$P_2 = (2, 5, 5)$



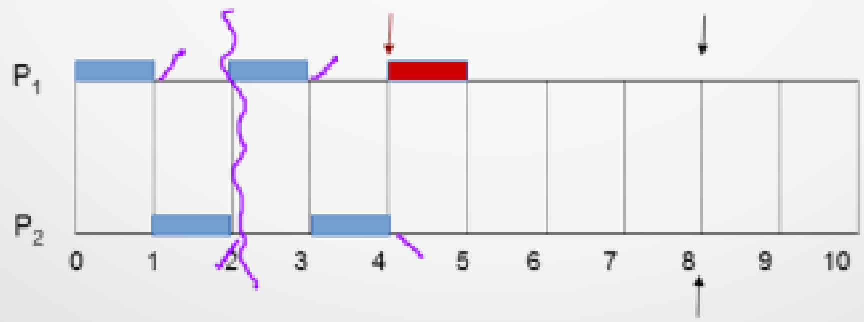
Slide10


Round-robin scheduling with preemption
P1 = (1,2,2)
P2 = (2,5,5)
P1
P2
IDLE
0
1
2
3
4
5
6
7
8
9
10

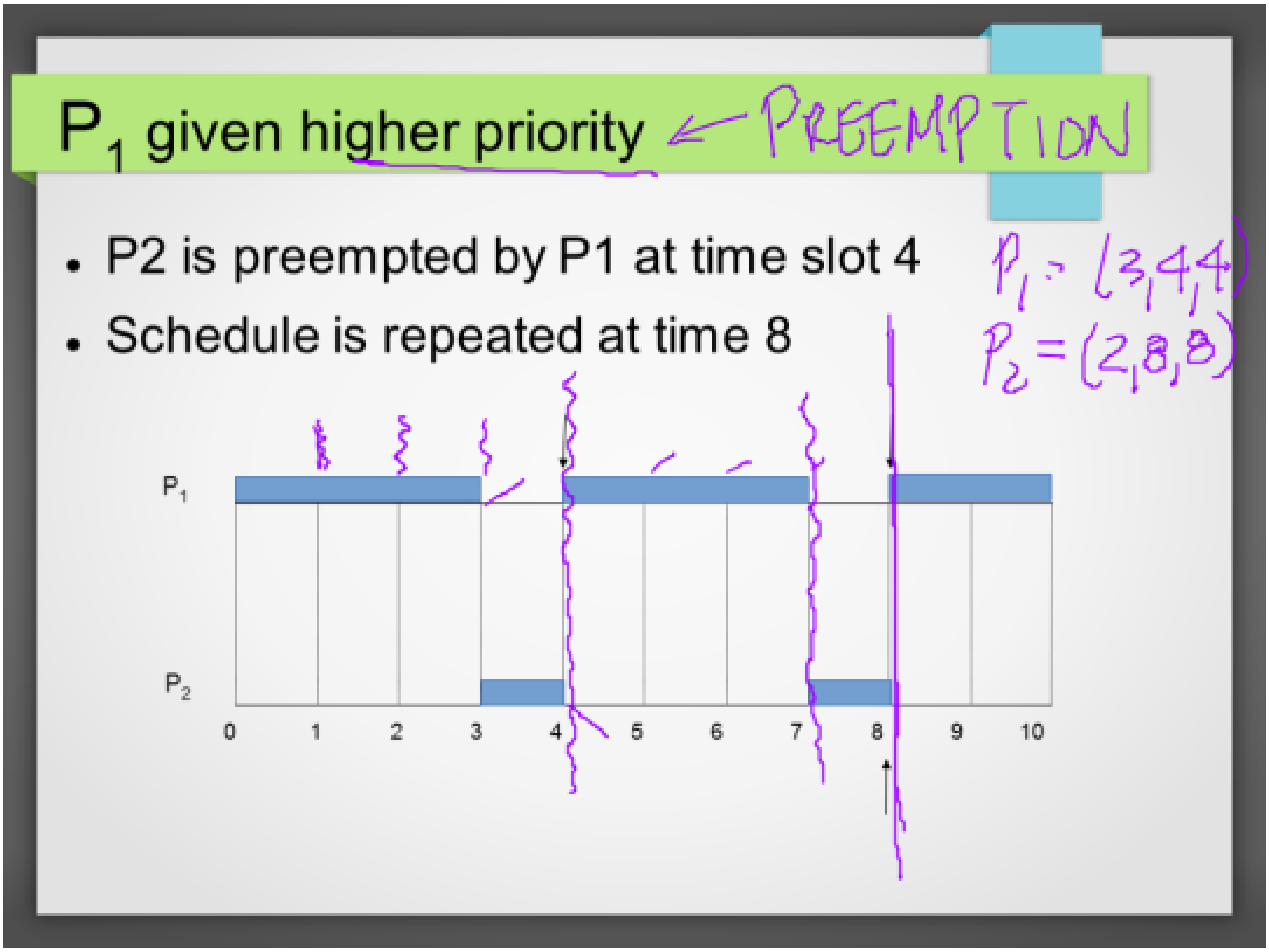
Slide11

Round-Robin Scheduling

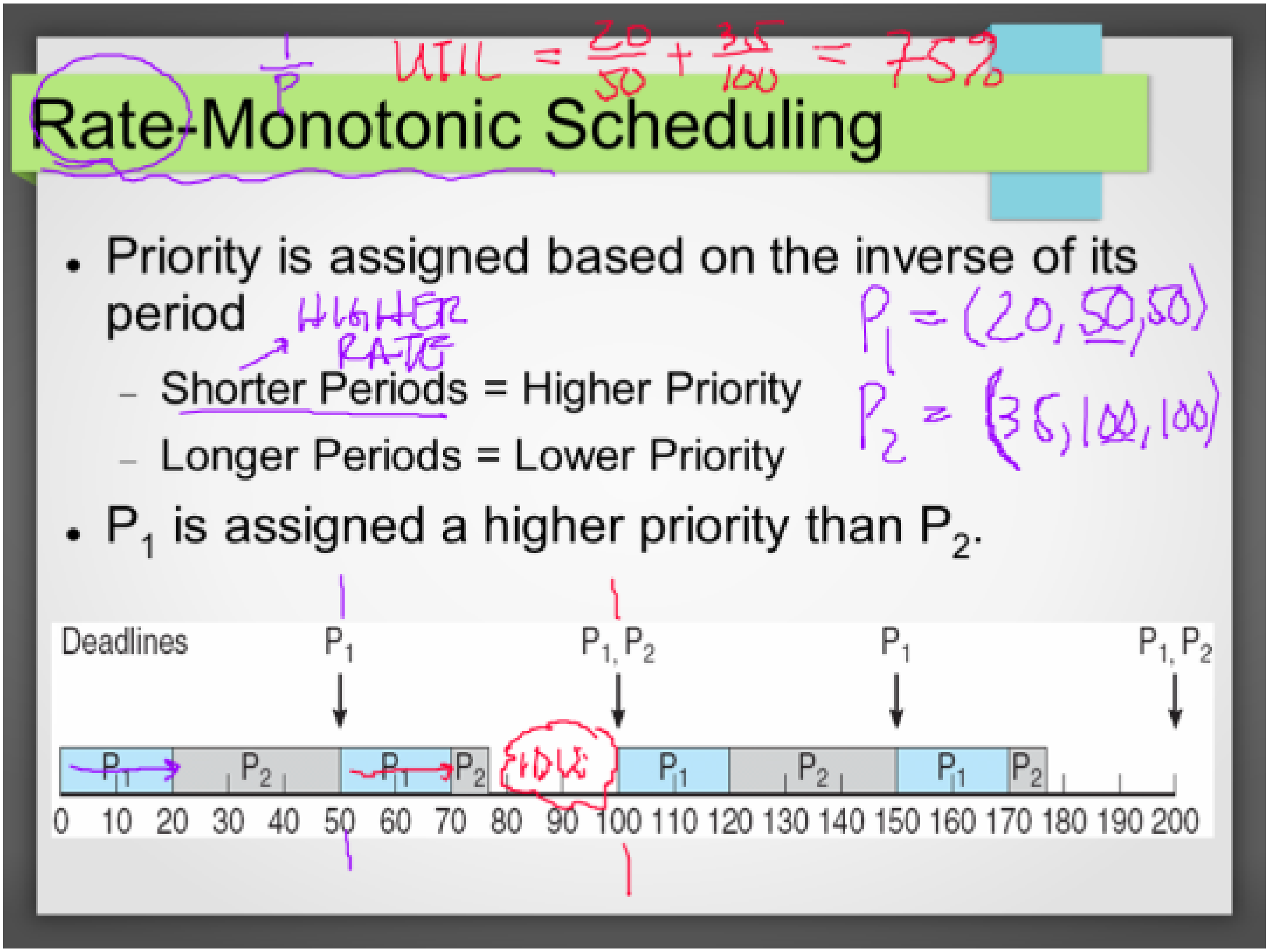
- Lets Assume 2 processes
  - $P_1 = (3,4,4)$
  - $P_2 = (2,8,8)$
  - Single Processor System



Slide12


P1 given higher priority
PREEMPTION
P2 is preempted by P1 at time slot 4
Schedule is repeated at time 8
P1 = (3,4,4)
P2 = (2,8,8)
P1
P2
0 1 2 3 4 5 6 7 8 9 10

Slide13



1/P
UTIL = 20/50 + 35/100 = 75%
Rate-Monotonic Scheduling
• Priority is assigned based on the inverse of its period
HIGHER RATE
– Shorter Periods = Higher Priority
– Longer Periods = Lower Priority
P1 = (20, 50, 50)
P2 = (35, 100, 100)
• P1 is assigned a higher priority than P2.
Deadlines
P1
P1, P2
P1
P1, P2
P1
P2
P1
P2
IDLE
P1
P2
P1
P2
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200

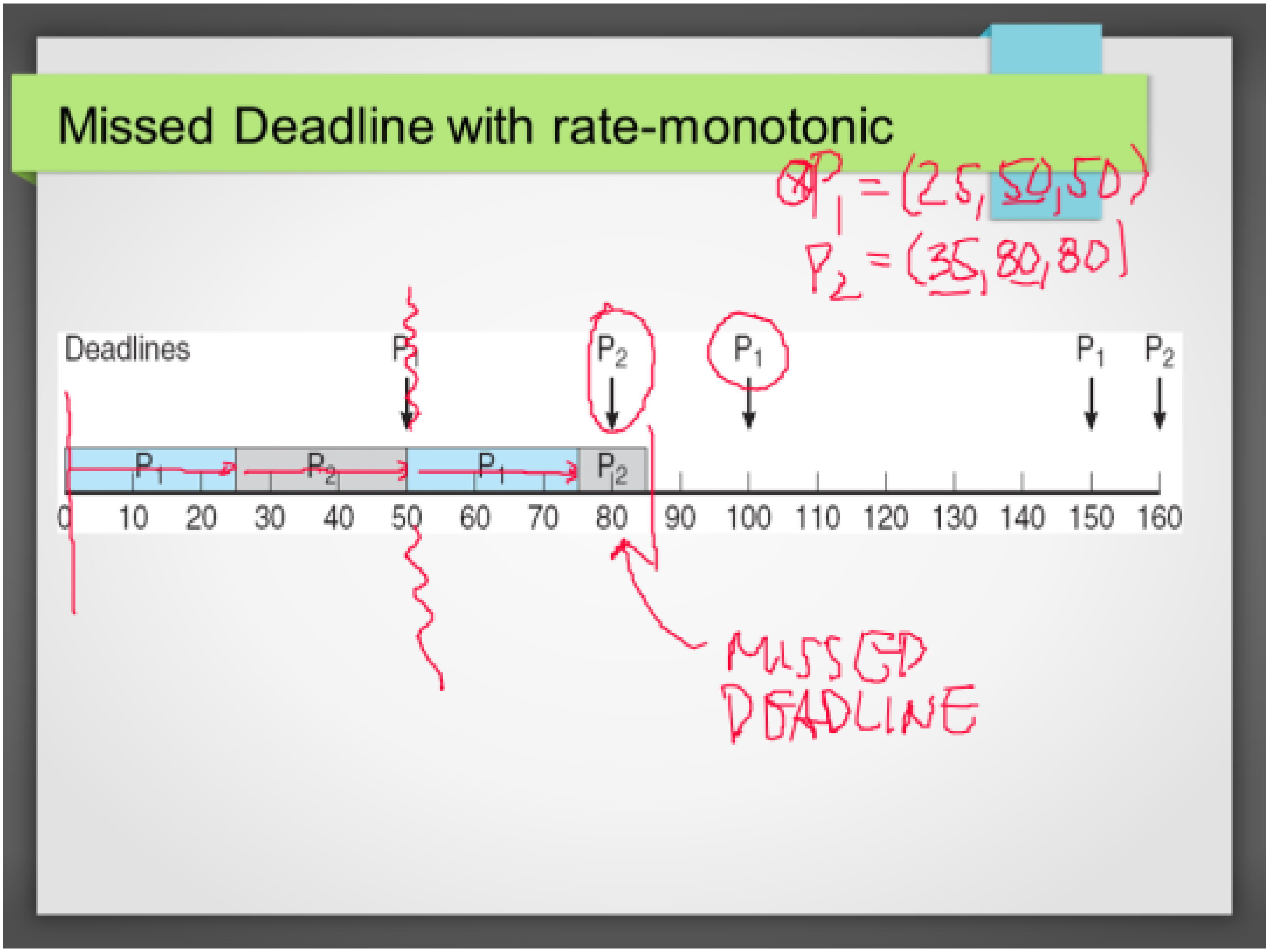
Slide14

Rate-Monotonic (cont.)

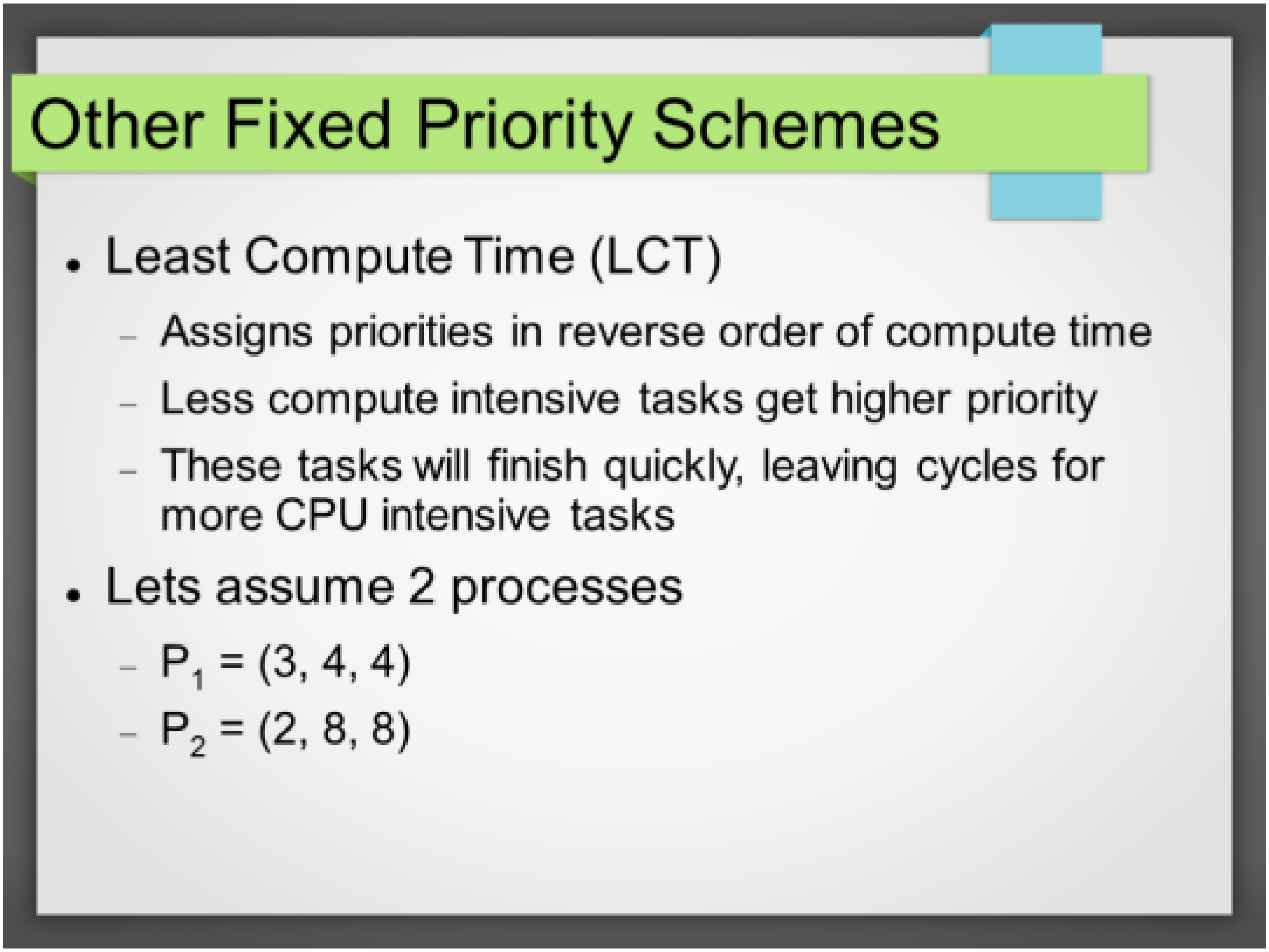
RMA

- Considered Optimal
  - If a set of processes cannot be scheduled by this algorithm, it cannot be scheduled by any other algorithm that assigns static priorities.
- Lets Assume 2 processes FIXED
  - $P_1 = (25, 50, 50)$
  - $P_2 = (35, 80, 80)$

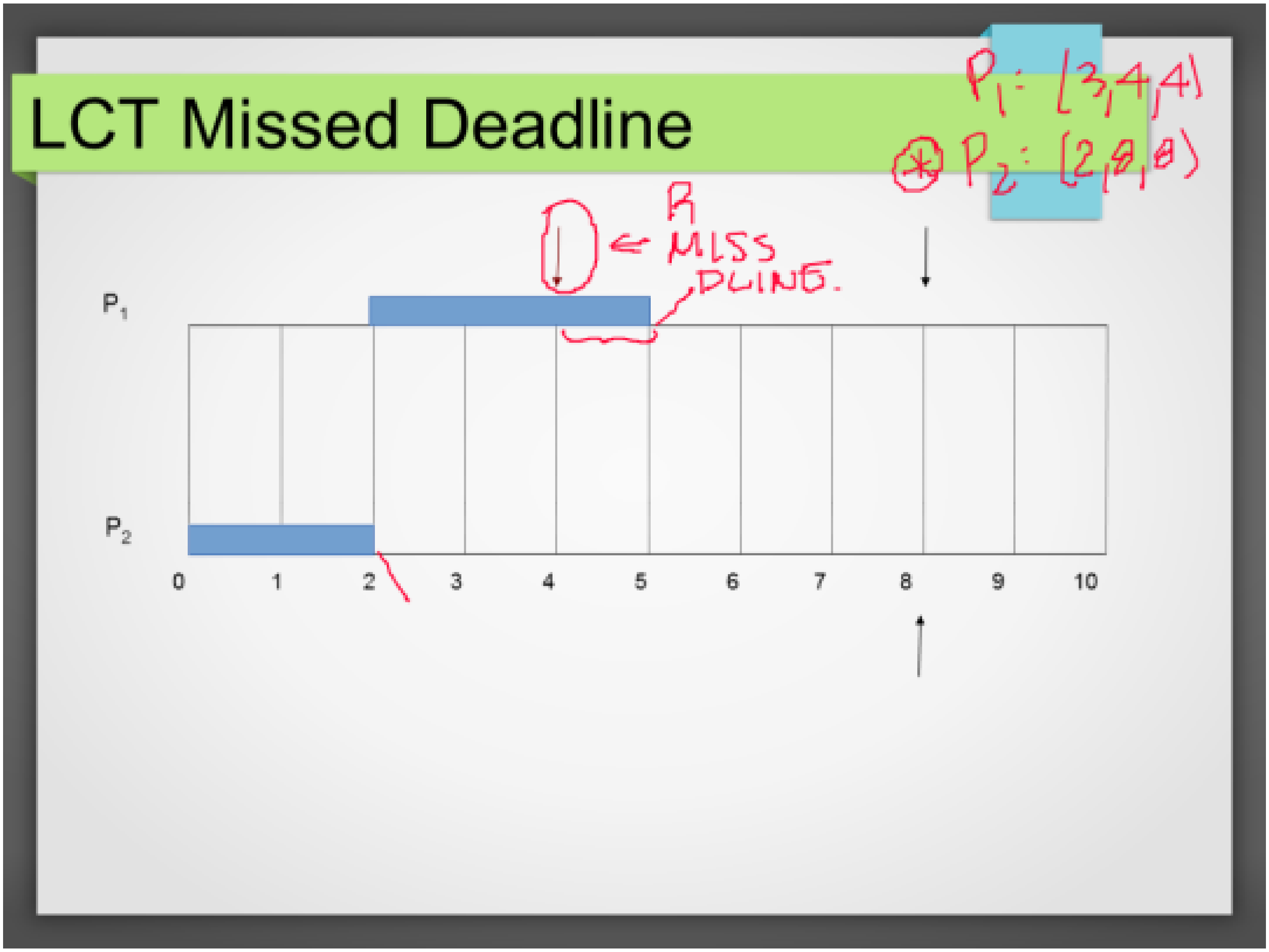
Slide15


Missed Deadline with rate-monotonic
P1 = (25, 50, 50)
P2 = (35, 80, 80)
Deadlines
P1
P2
P1
P1
P2
P1
P2
P1
P2
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
MISSED
DEADLINE

Slide16


Other Fixed Priority Schemes
• Least Compute Time (LCT)
– Assigns priorities in reverse order of compute time
– Less compute intensive tasks get higher priority
– These tasks will finish quickly, leaving cycles for more CPU intensive tasks
• Lets assume 2 processes
– $P_1 = (3, 4, 4)$
– $P_2 = (2, 8, 8)$

Slide17

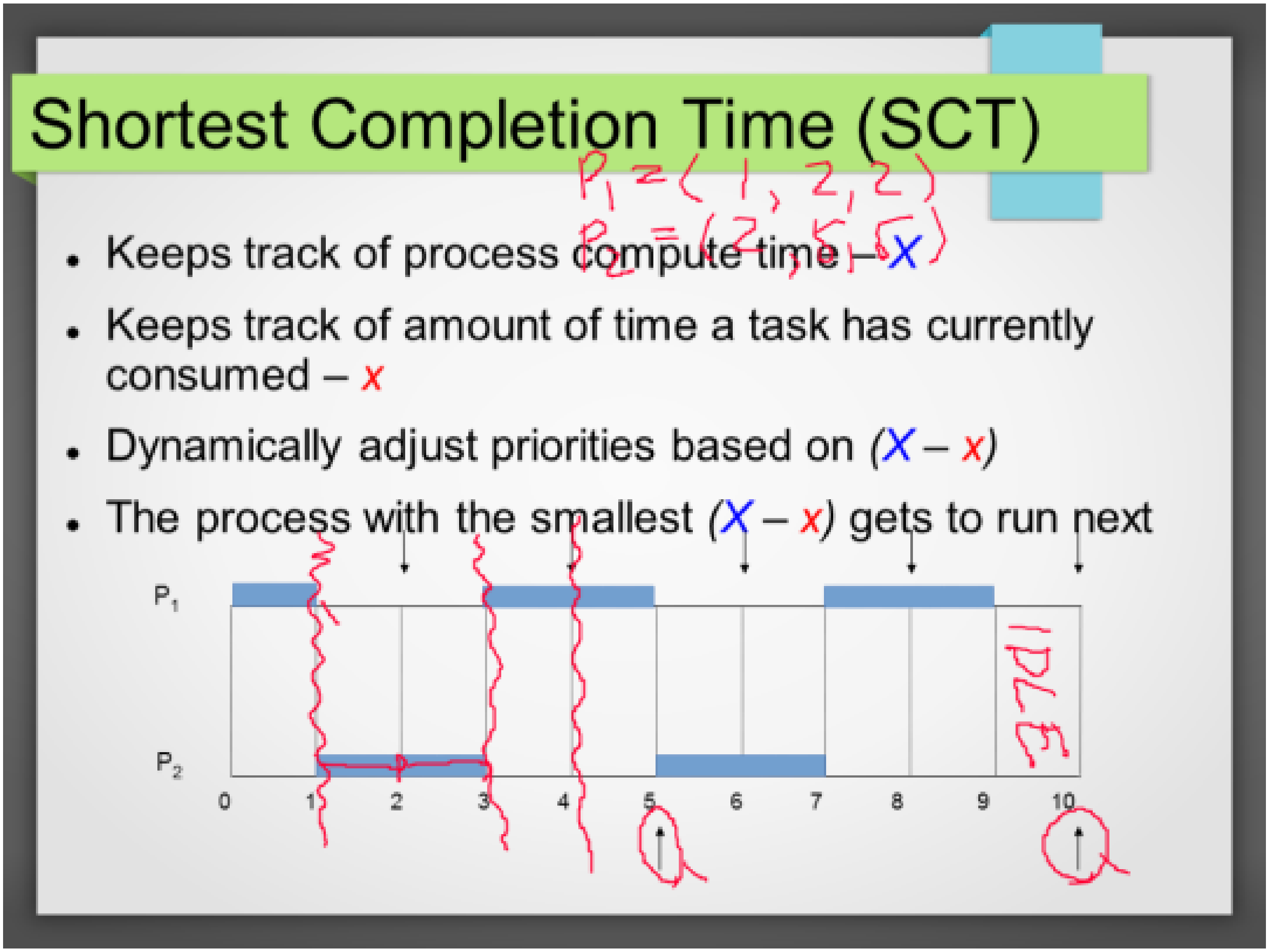

LCT Missed Deadline
P1: [3,4,4]
P2: [2,8,8]
R
MISS
DLINE.
P1
P2
0
1
2
3
4
5
6
7
8
9
10

Slide18

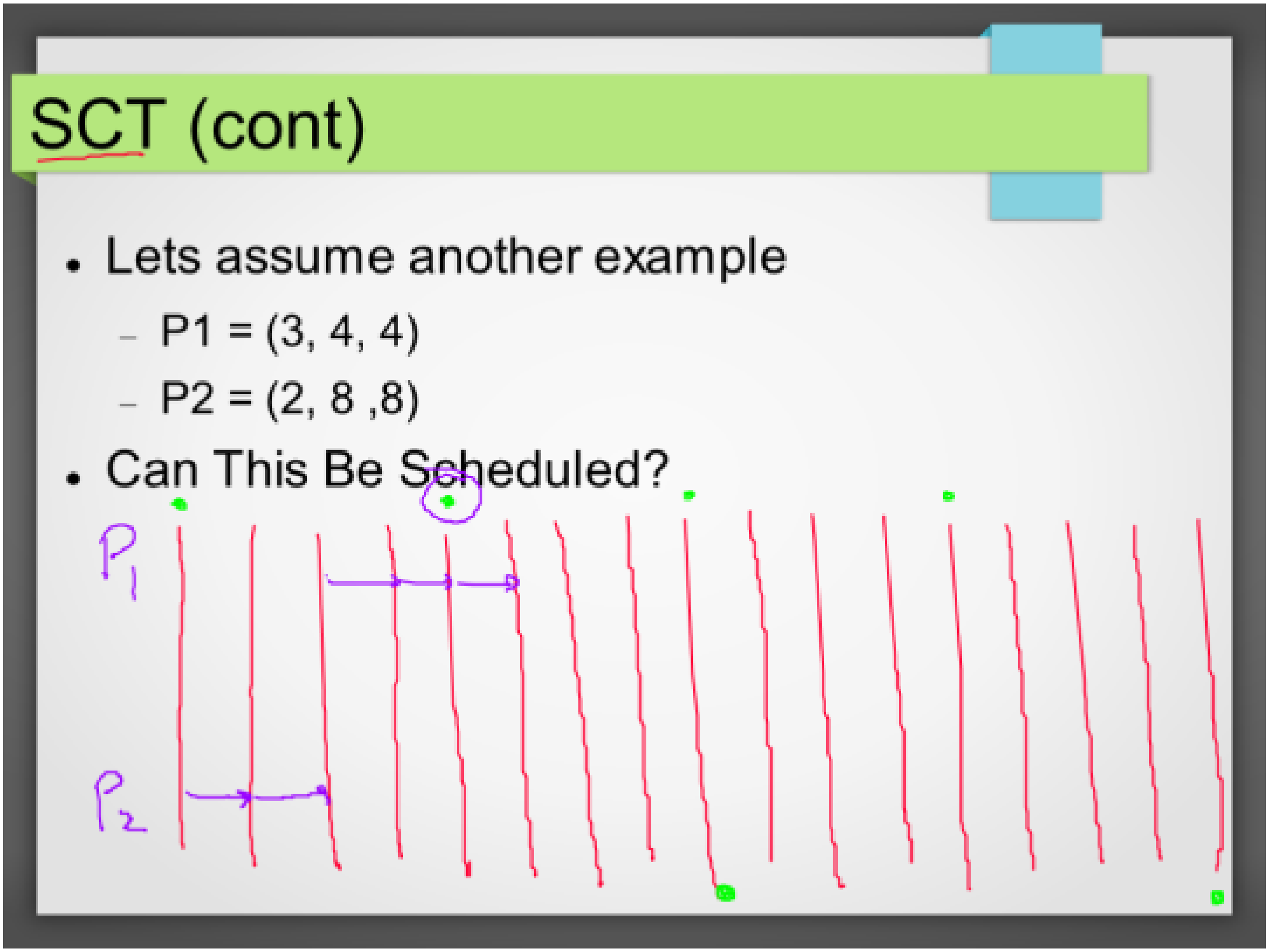
Dynamic Priority

- Dynamic Priority
  - Priority of a task will change dependent on scheduler logic throughout the execution cycle of the system
    - Can be a combination of static and dynamic where priorities are fixed except for some well-defined places that require dynamic changes.
- Three Common Dynamic schemes
  - Shortest Completion Time (SCT)
  - Earliest Deadline First (EDF)
  - Least Slack Time (LST)

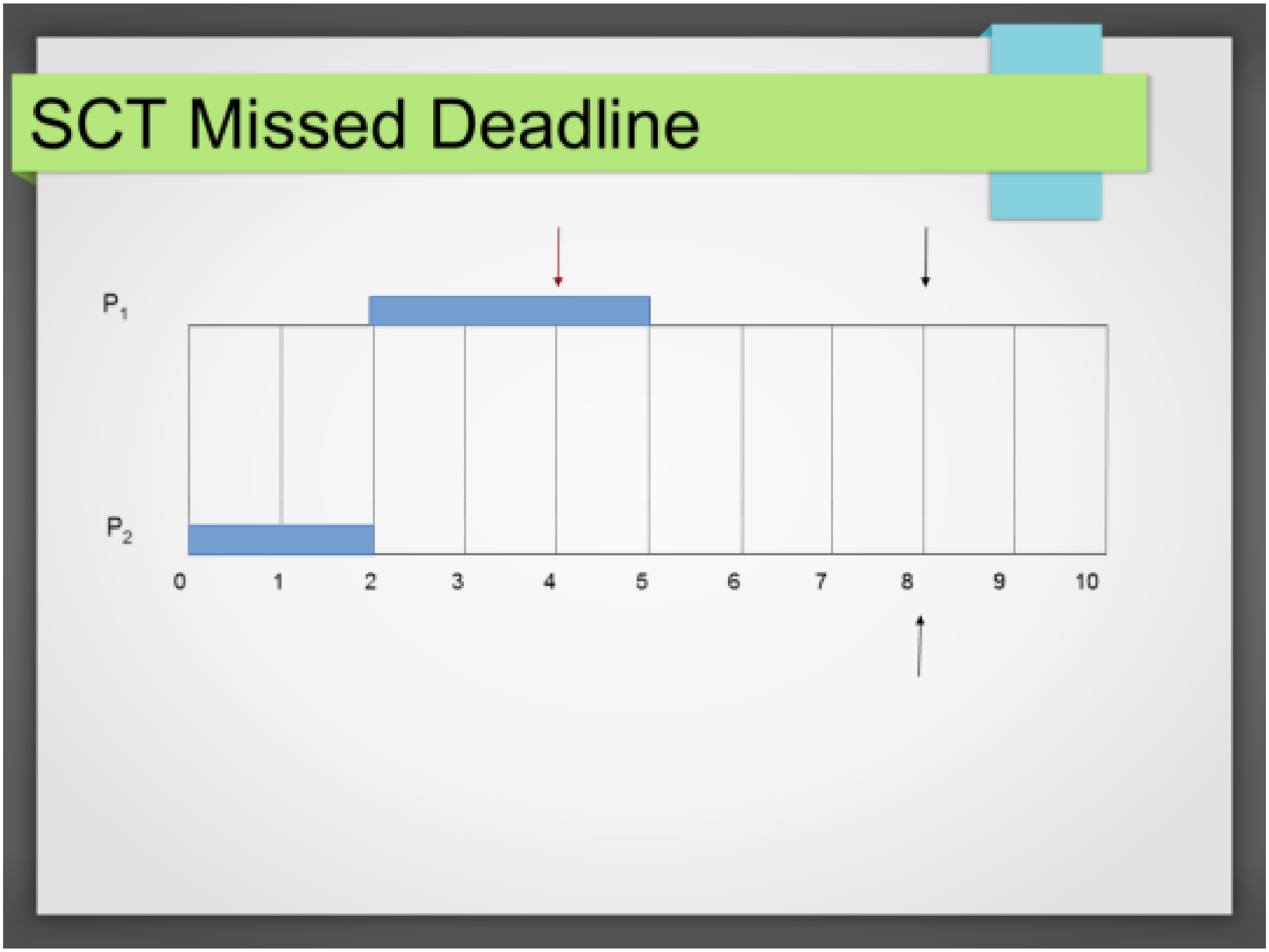
Slide19


Shortest Completion Time (SCT)
P1 = (1, 2, 2)
P2 = (2, 5, 5)
Keeps track of process compute time – X
Keeps track of amount of time a task has currently consumed – x
Dynamically adjust priorities based on (X – x)
The process with the smallest (X – x) gets to run next
P1
P2
IDLE
0
1
2
3
4
5
6
7
8
9
10

Slide20


SCT (cont)
Lets assume another example
P1 = (3, 4, 4)
P2 = (2, 8 ,8)
Can This Be Scheduled?
P1
P2

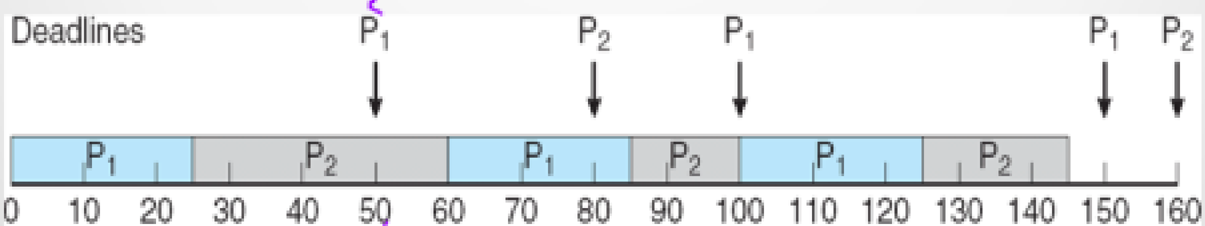
Slide21



Slide22

Earliest Deadline-First (EDF) Scheduling

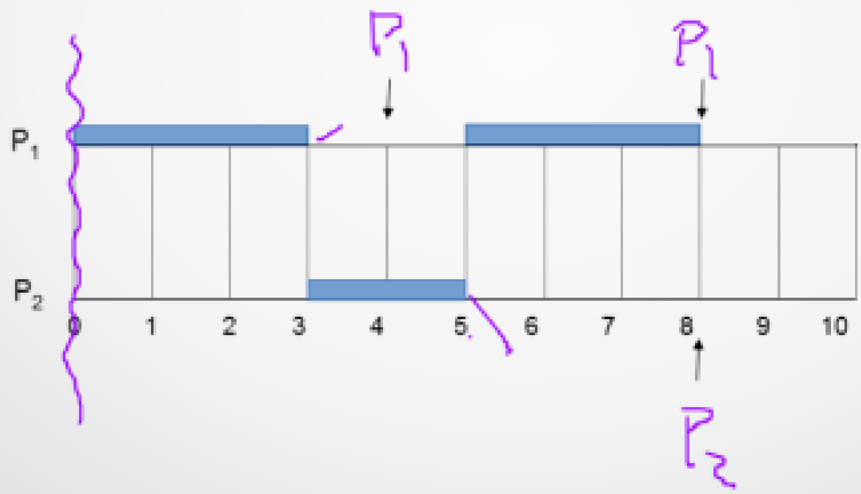
- Priorities are assigned according to deadline
  - Earlier the deadline, the higher the priority
  - Later the deadline, the lower the priority
- When a process gets to the ready state it must
  - Announce its deadline requirements
  - Priorities may have to be adjusted to deal with the new process



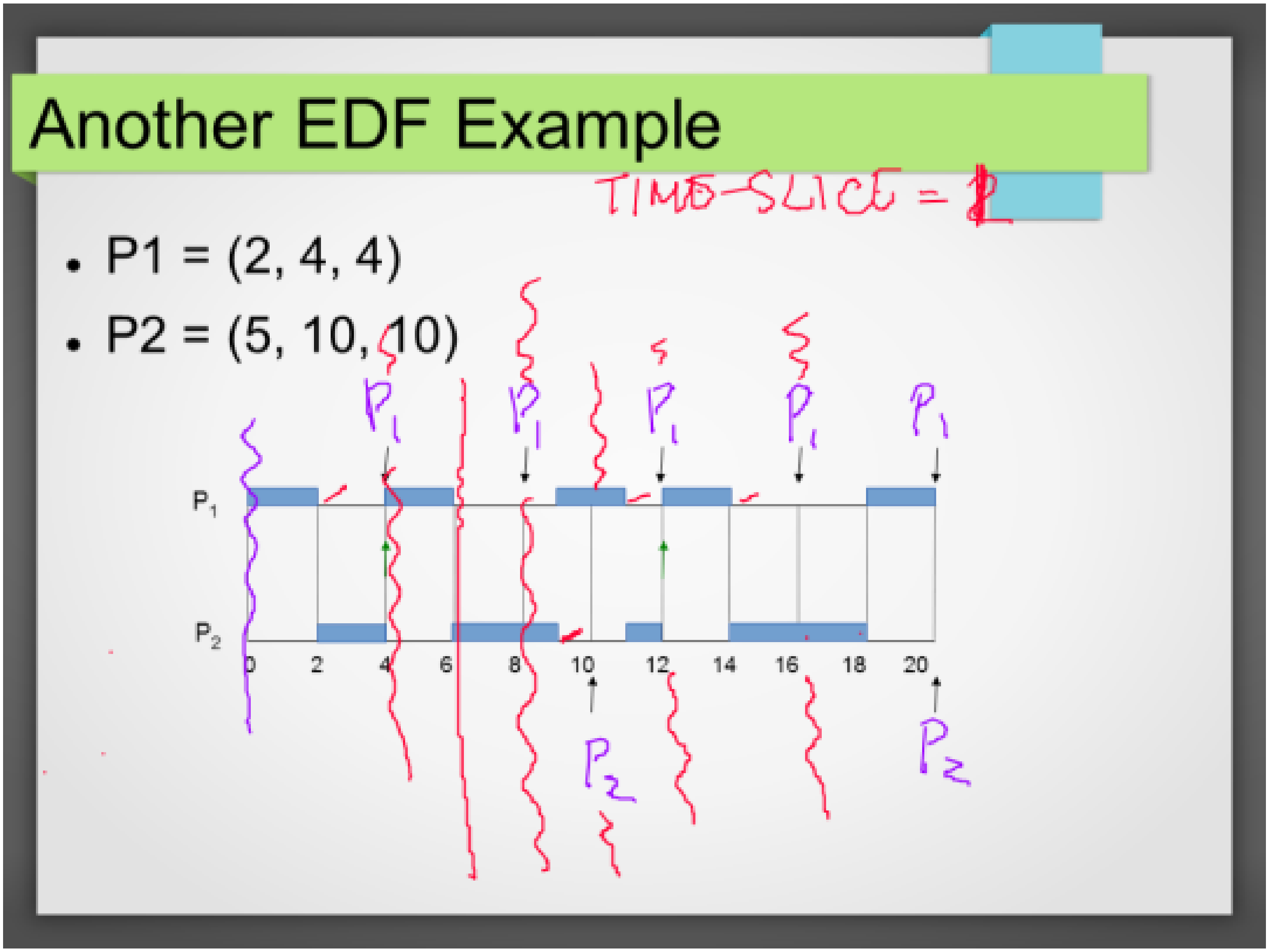
Slide23

EDF Scheduling

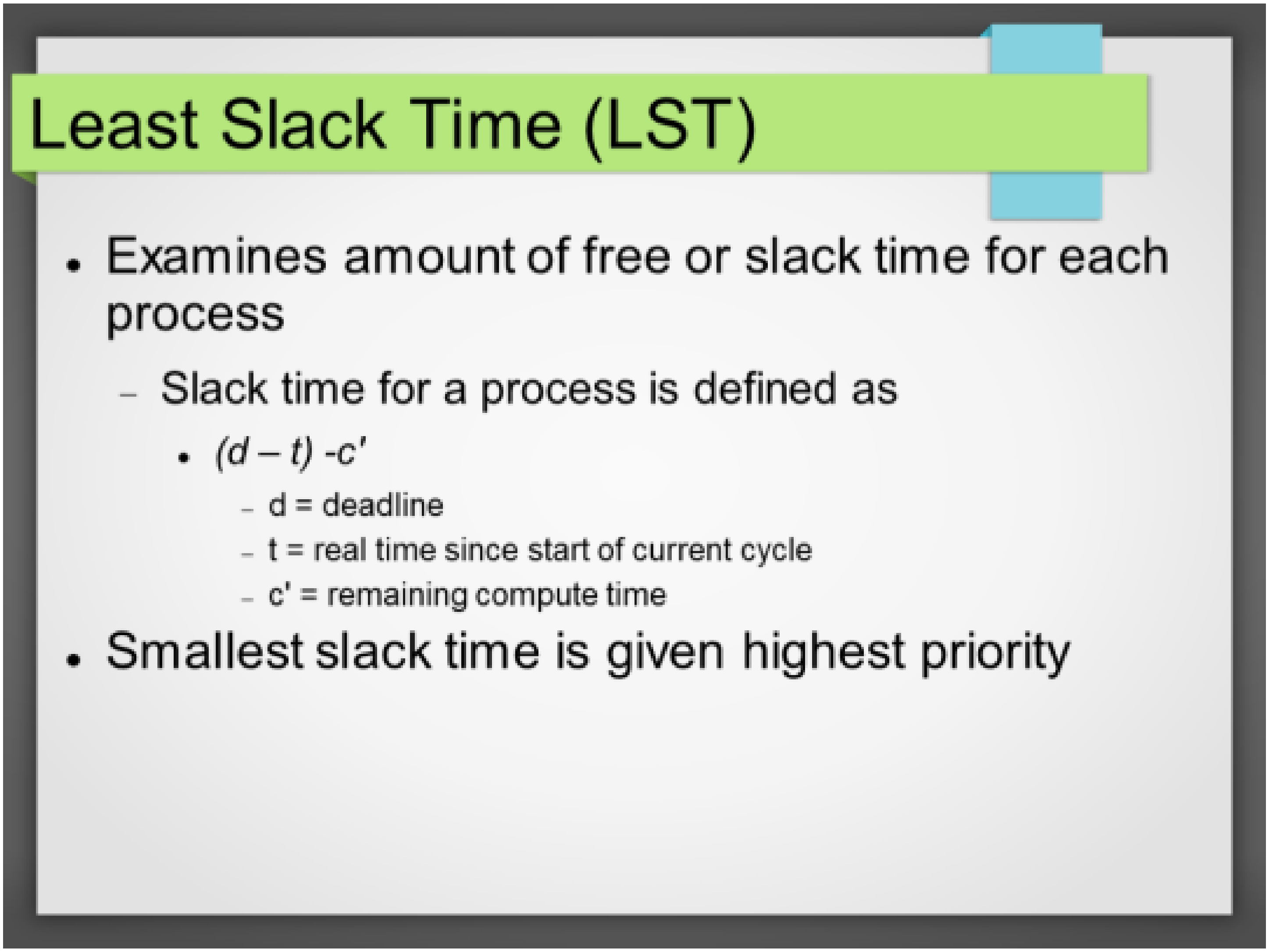
- Lets look at our old friend
  - $P_1 = (3, 4, 4)$
  - $P_2 = (2, 8, 8)$



Slide24


Another EDF Example
TIME-SLICE = 1
P1 = (2, 4, 4)
P2 = (5, 10, 10)
P1
P2
0 2 4 6 8 10 12 14 16 18 20

Slide25

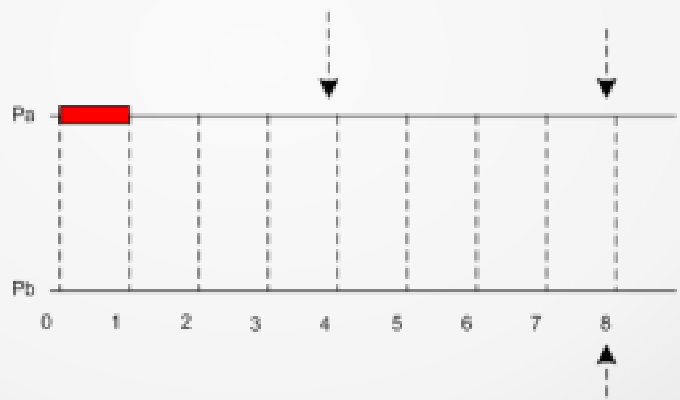

Least Slack Time (LST)
Examines amount of free or slack time for each process
Slack time for a process is defined as
(d – t) -c'
d = deadline
t = real time since start of current cycle
c' = remaining compute time
Smallest slack time is given highest priority

Slide26

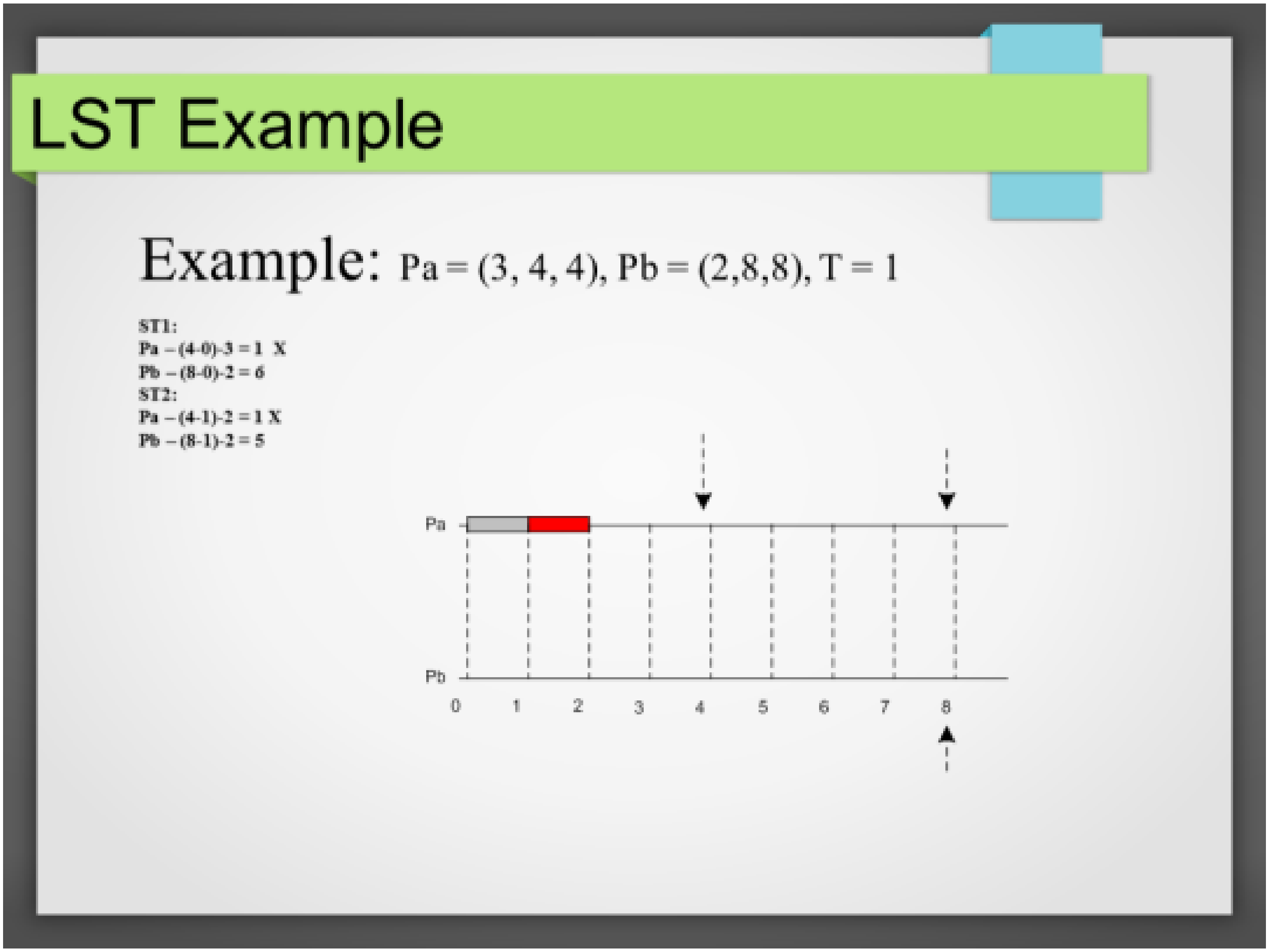
LST Example

Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1

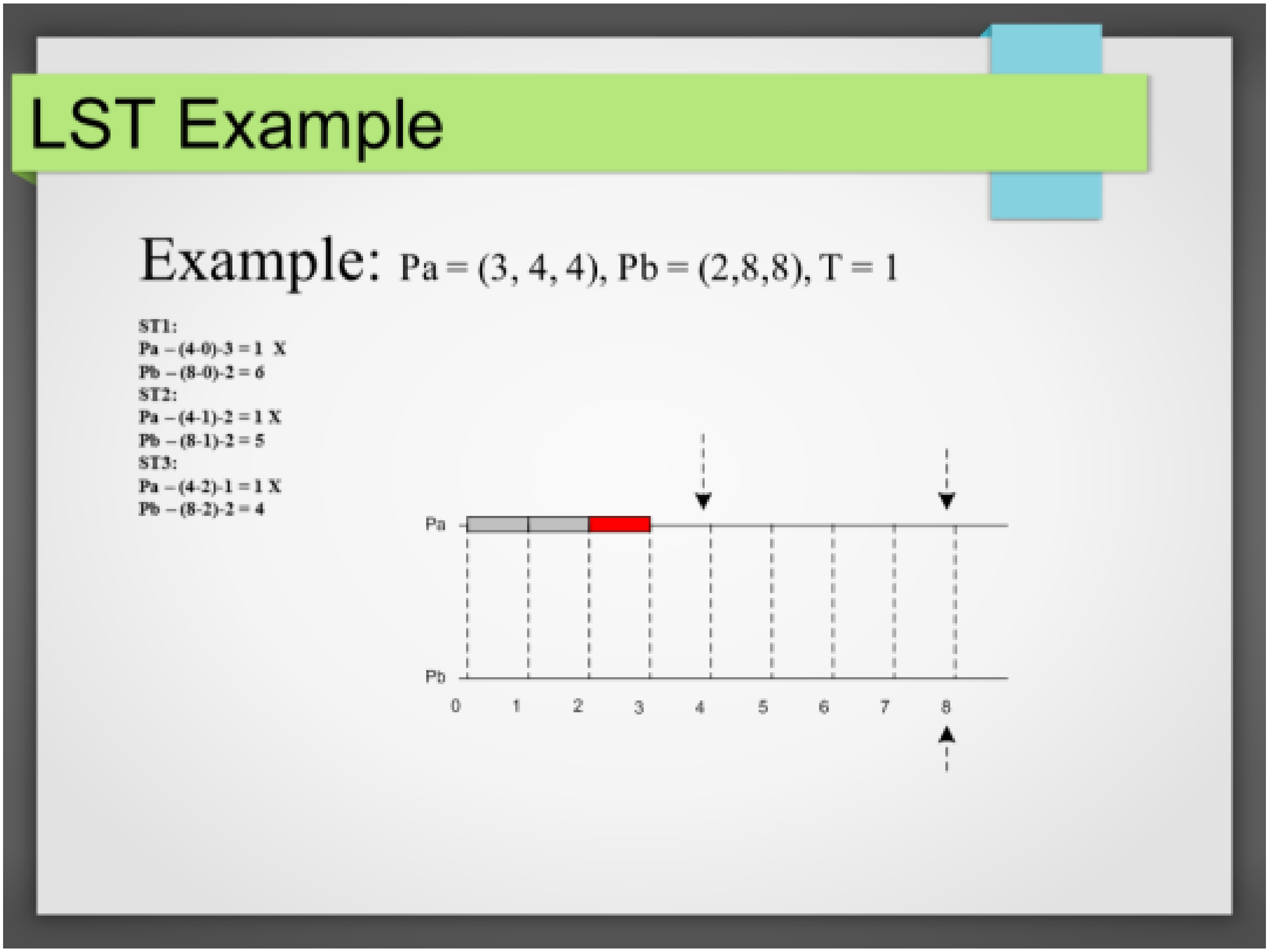
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6



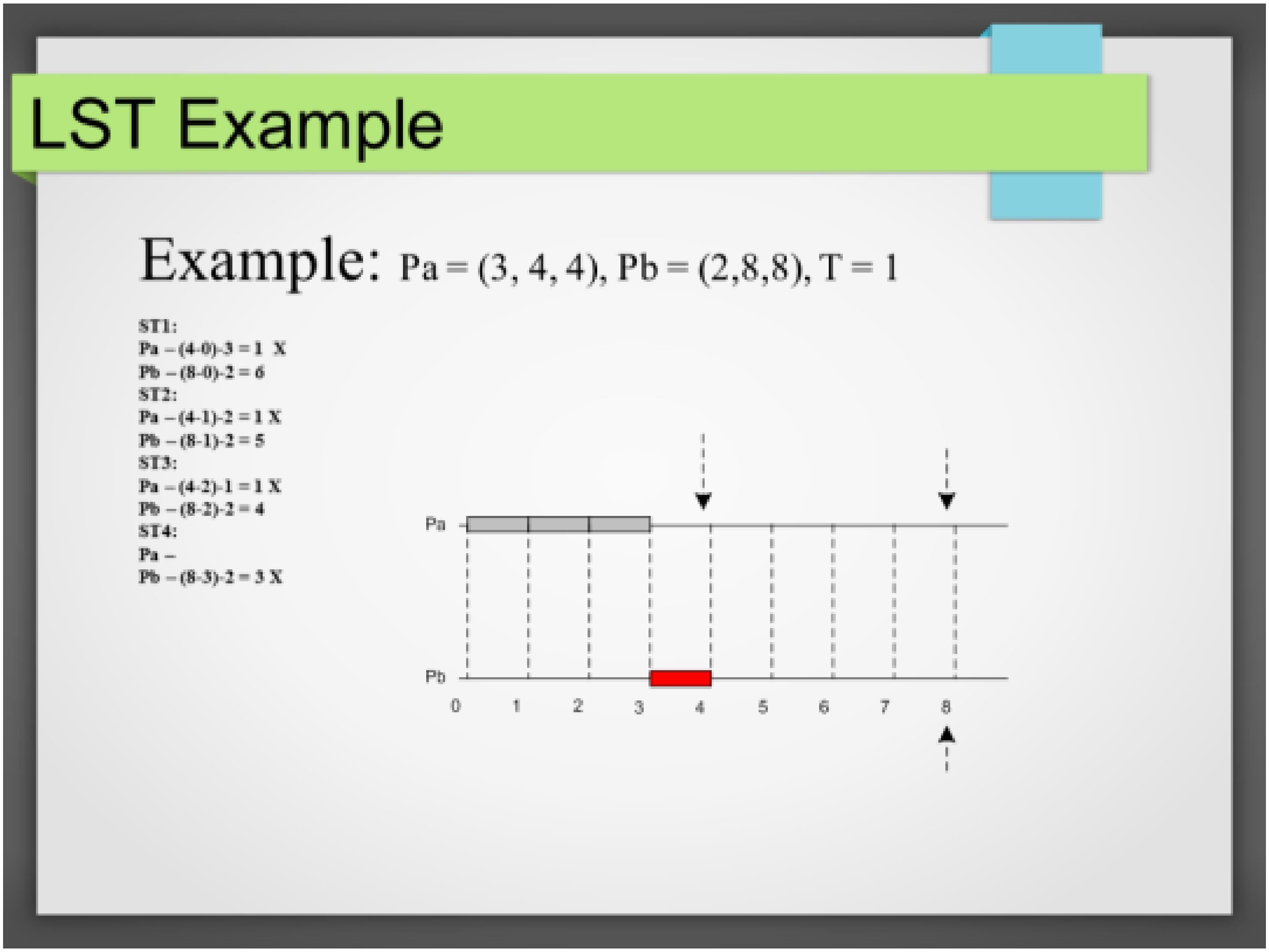
Slide27


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
Pa
Pb
0
1
2
3
4
5
6
7
8

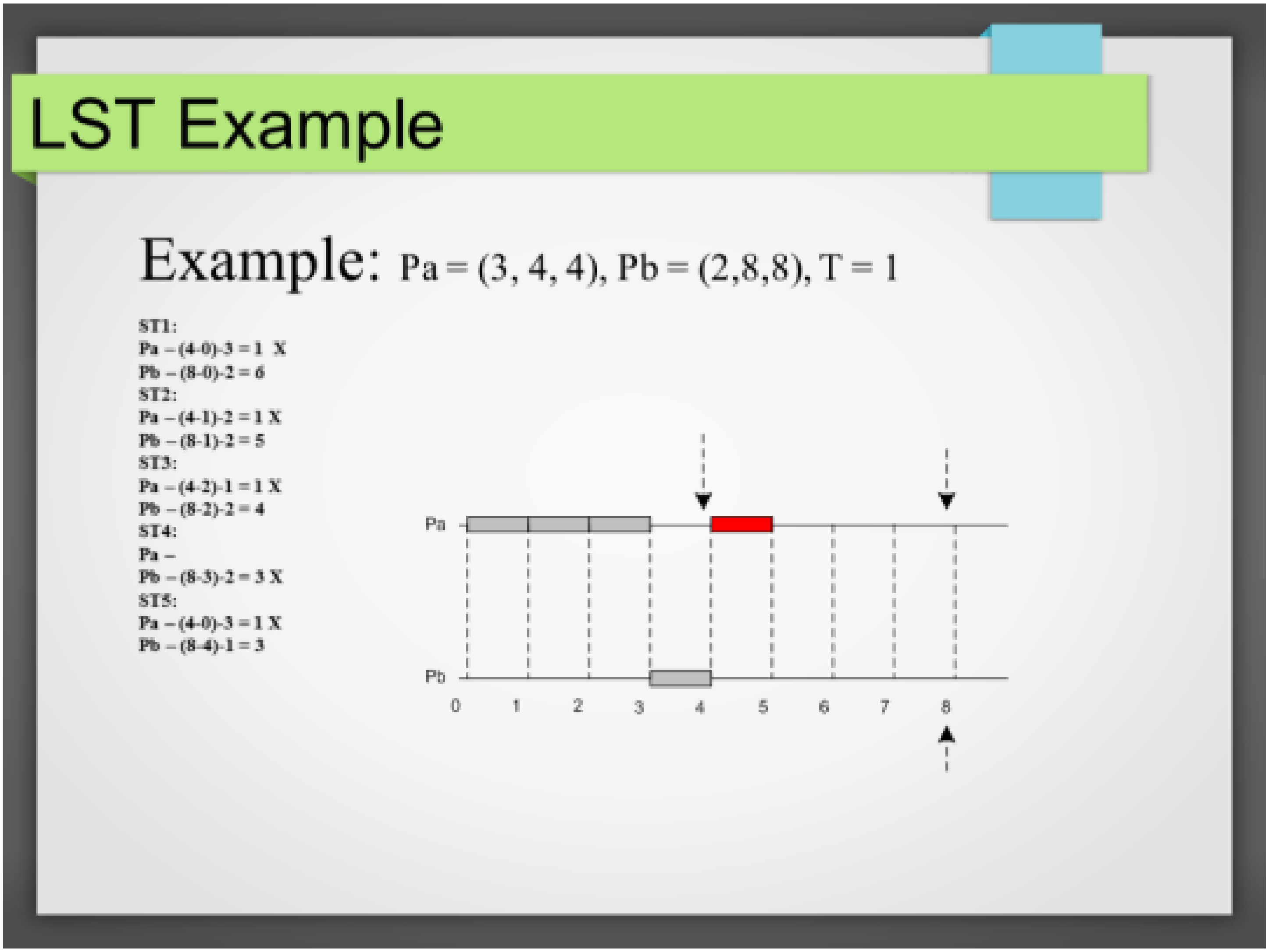
Slide28


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
ST3:
Pa – (4-2)-1 = 1 X
Pb – (8-2)-2 = 4
Pa
Pb
0 1 2 3 4 5 6 7 8

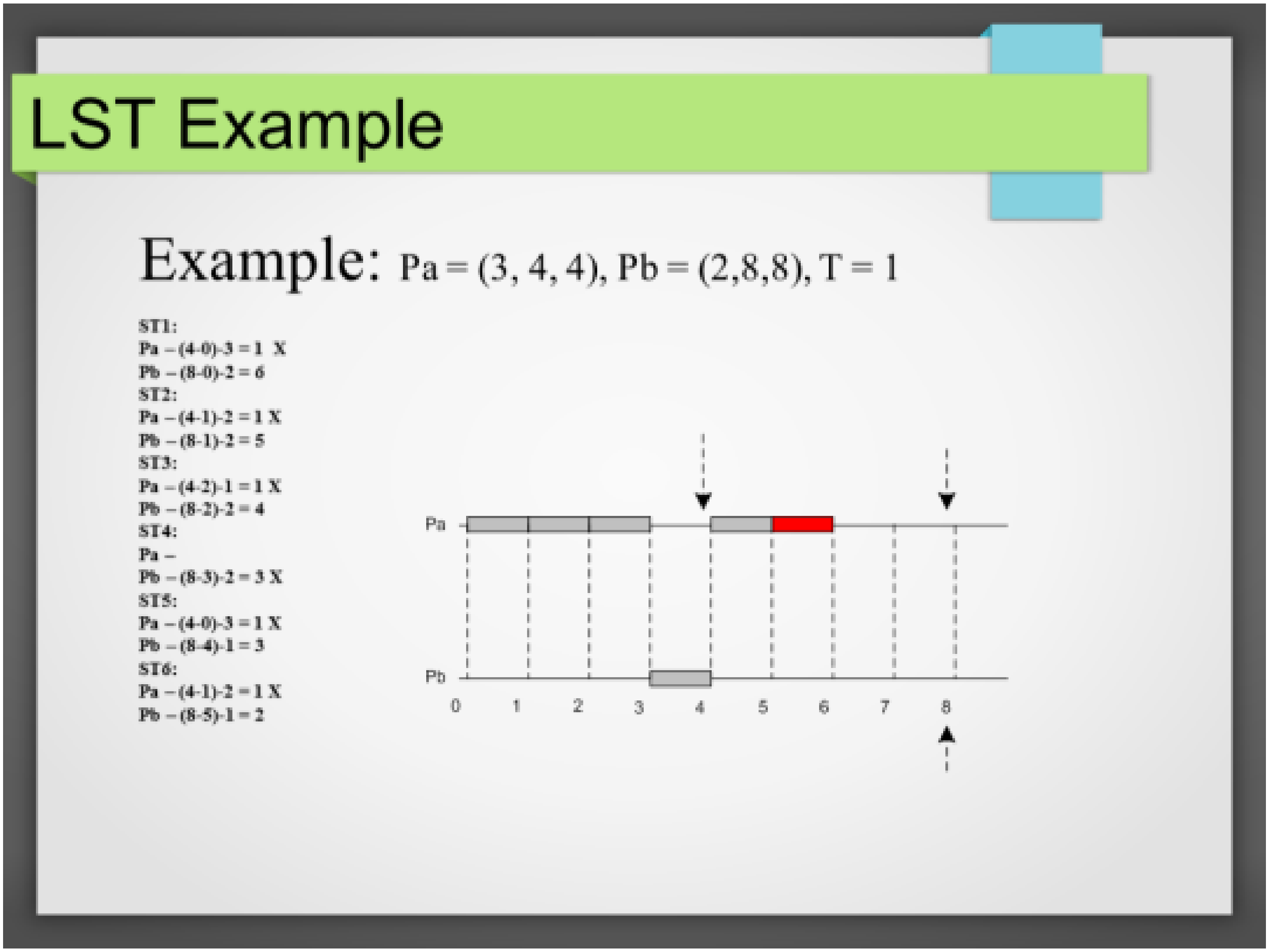
Slide29


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
ST3:
Pa – (4-2)-1 = 1 X
Pb – (8-2)-2 = 4
ST4:
Pa –
Pb – (8-3)-2 = 3 X
Pa
Pb
0 1 2 3 4 5 6 7 8

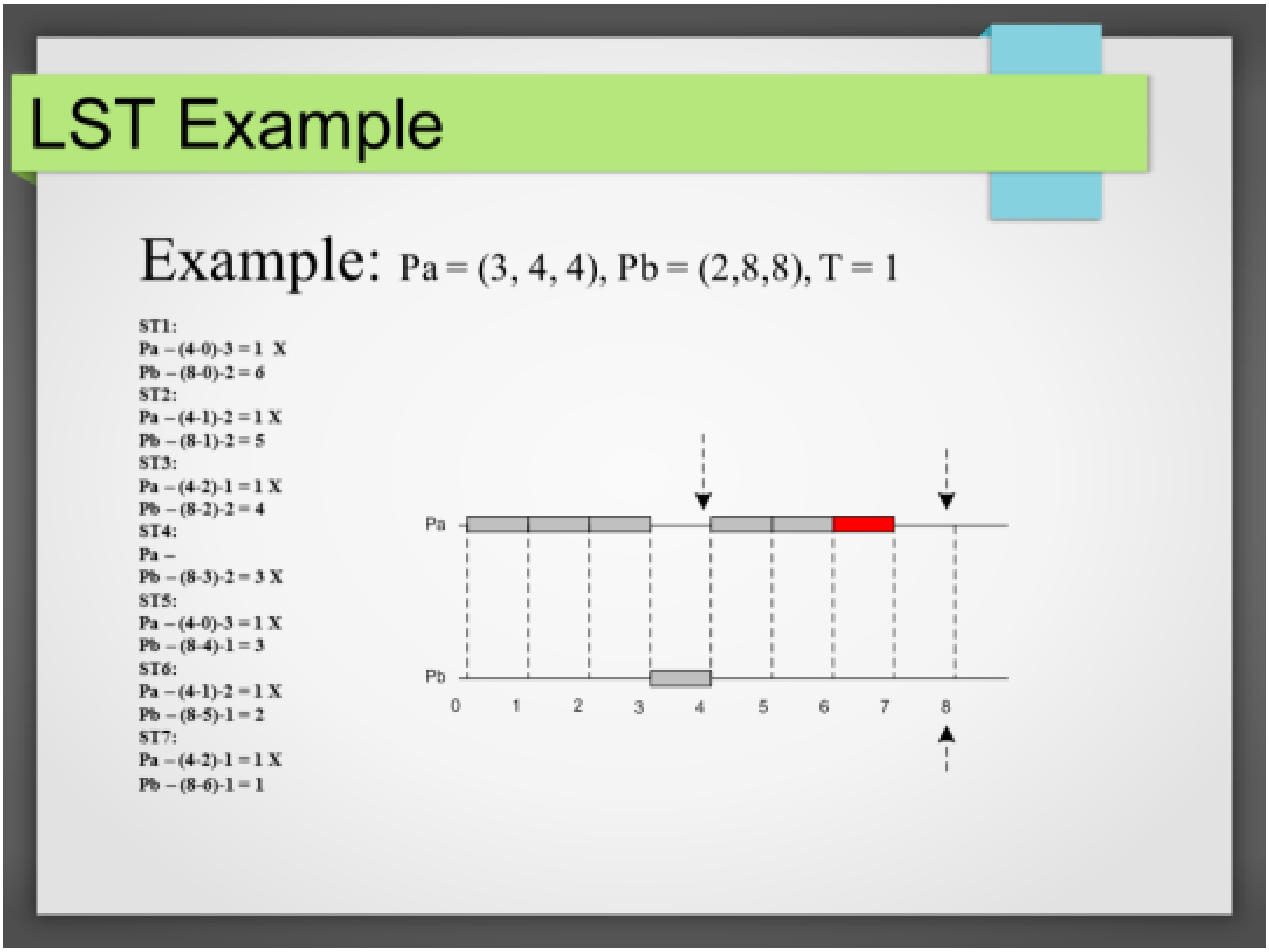
Slide30


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
ST3:
Pa – (4-2)-1 = 1 X
Pb – (8-2)-2 = 4
ST4:
Pa –
Pb – (8-3)-2 = 3 X
ST5:
Pa – (4-0)-3 = 1 X
Pb – (8-4)-1 = 3
Pa
Pb
0 1 2 3 4 5 6 7 8

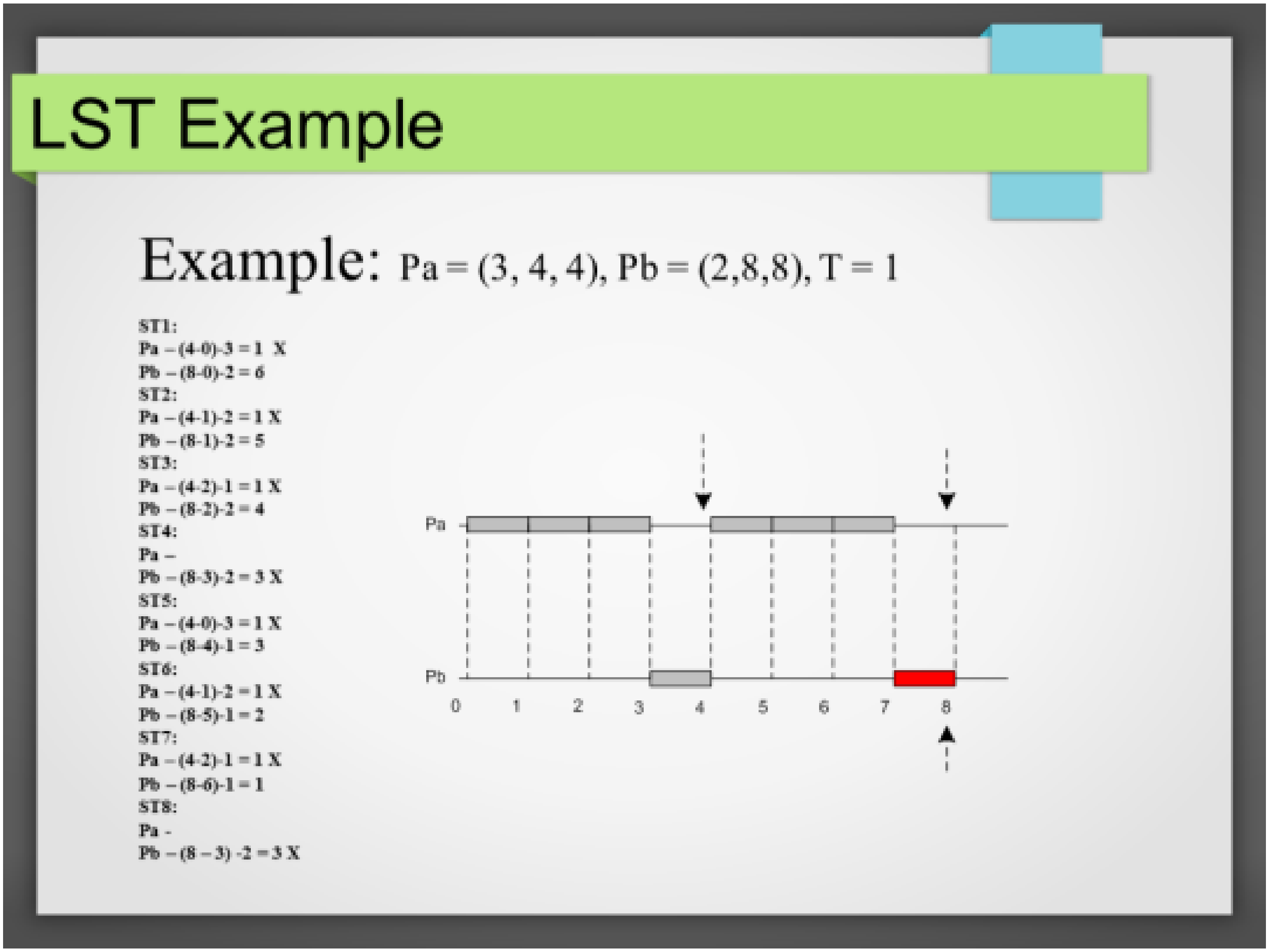
Slide31


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
ST3:
Pa – (4-2)-1 = 1 X
Pb – (8-2)-2 = 4
ST4:
Pa –
Pb – (8-3)-2 = 3 X
ST5:
Pa – (4-0)-3 = 1 X
Pb – (8-4)-1 = 3
ST6:
Pa – (4-1)-2 = 1 X
Pb – (8-5)-1 = 2
Pa
Pb
0
1
2
3
4
5
6
7
8

Slide32


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
ST3:
Pa – (4-2)-1 = 1 X
Pb – (8-2)-2 = 4
ST4:
Pa –
Pb – (8-3)-2 = 3 X
ST5:
Pa – (4-0)-3 = 1 X
Pb – (8-4)-1 = 3
ST6:
Pa – (4-1)-2 = 1 X
Pb – (8-5)-1 = 2
ST7:
Pa – (4-2)-1 = 1 X
Pb – (8-6)-1 = 1
Pa
Pb
0
1
2
3
4
5
6
7
8

Slide33


LST Example
Example: Pa = (3, 4, 4), Pb = (2,8,8), T = 1
ST1:
Pa – (4-0)-3 = 1 X
Pb – (8-0)-2 = 6
ST2:
Pa – (4-1)-2 = 1 X
Pb – (8-1)-2 = 5
ST3:
Pa – (4-2)-1 = 1 X
Pb – (8-2)-2 = 4
ST4:
Pa –
Pb – (8-3)-2 = 3 X
ST5:
Pa – (4-0)-3 = 1 X
Pb – (8-4)-1 = 3
ST6:
Pa – (4-1)-2 = 1 X
Pb – (8-5)-1 = 2
ST7:
Pa – (4-2)-1 = 1 X
Pb – (8-6)-1 = 1
ST8:
Pa -
Pb – (8 – 3) -2 = 3 X
Pa
Pb
0
1
2
3
4
5
6
7
8