Course Name: Analysis and Design of Algorithms

Topics to be covered

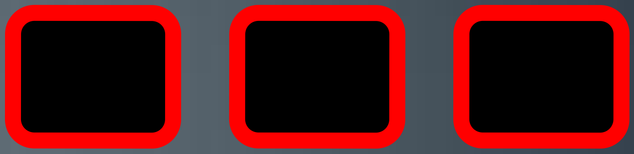
- Branch and Bound
  - Job Scheduling
  - Comparing with Greedy Approach

Branch and Bound Algorithm:

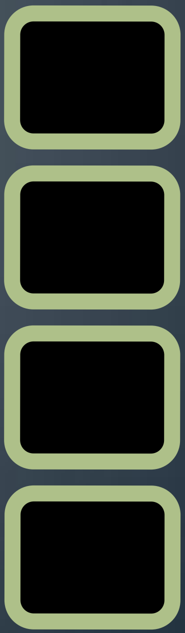
Scheduling Problem

Material by A.Mirhashemi

Input of the problem:



- A number of resources
- A number of tasks



Output of the problem:

- A sequence of feeding the tasks to resources

to minimize the required processing time

Application 1

Digital processing:

Each resource is a processor. All tasks need to pass trough all processors in the fix sequence A,B,C but depending on the task it takes different time for each processor to process them. For example :

Processor A:	Scanning
Processor B:	Making a PDF
Processor C:	Exporting a PDF

Task 1:	A one page plain text document
Task 2:	A 10 page document with pictures
Task 3:	A 5 page html document.
Task 4:	…

Application 2

Production line:

Each product (task) need to pass trough all machines (resources) in the production line but, the time depends on what kind of customization the customer has ordered for that production. For example:

Machine A:	Solding
Machine B:	Painting
Machine C:	Packaging

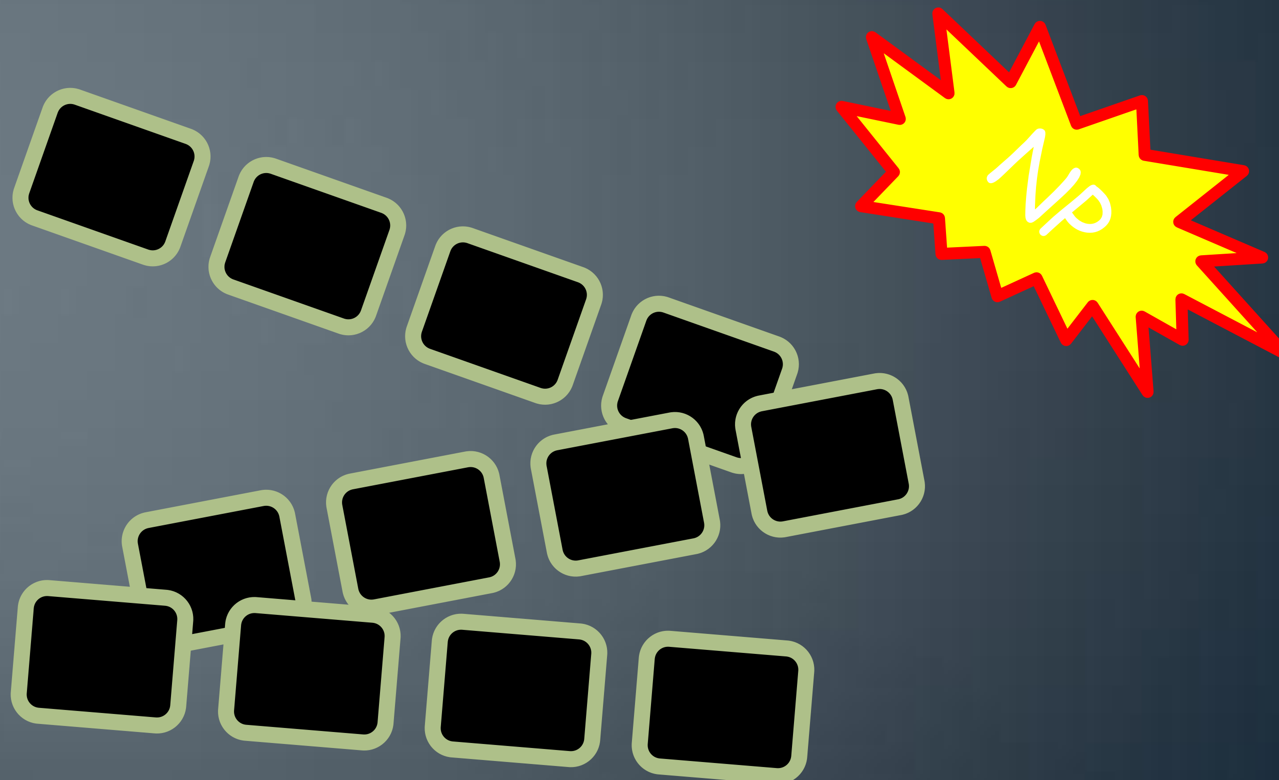
Task 1:	A black car with airbag
Task 2:	A red car without airbag with CD player
Task 3:	A white car with leather seats
Task 4:	…

Different tasks take different time to be processed in each resource

7	6	7
5	5	2
6	4	1
3	4	3

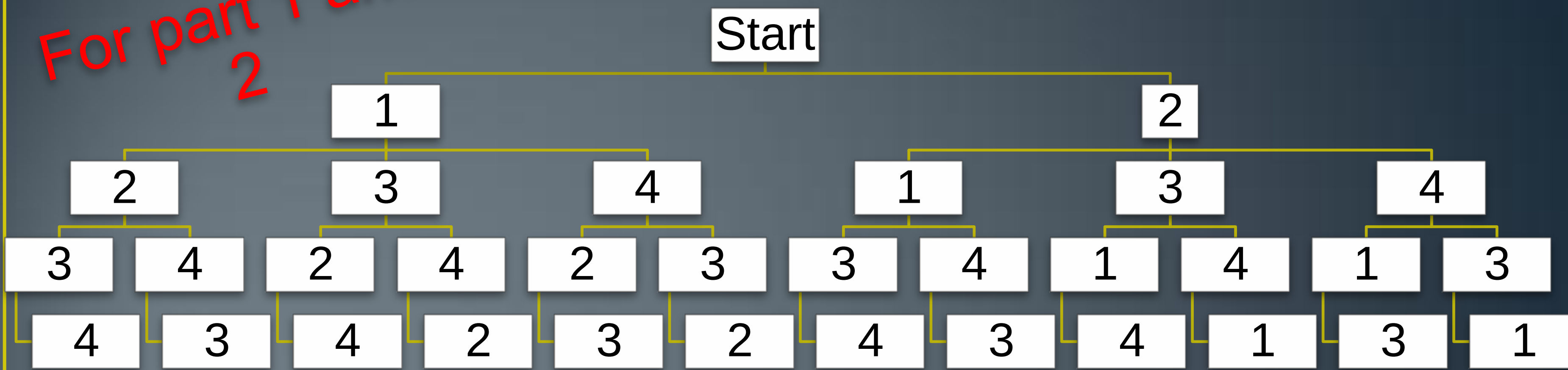
Tasks can be done in any order

N! Possible different sequences

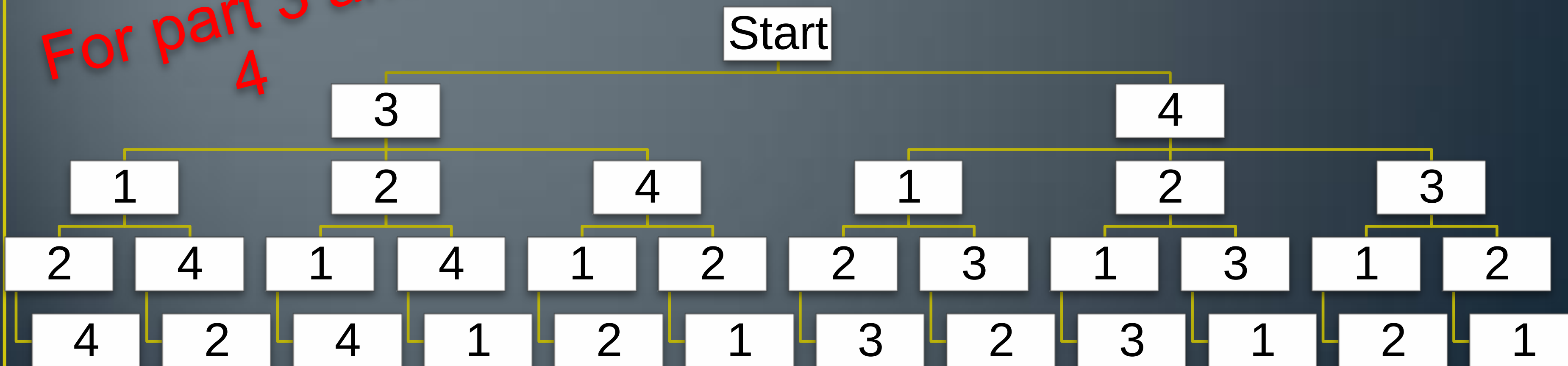


Decision tree (Brute force)

For part 1 and 2



For part 3 and 4



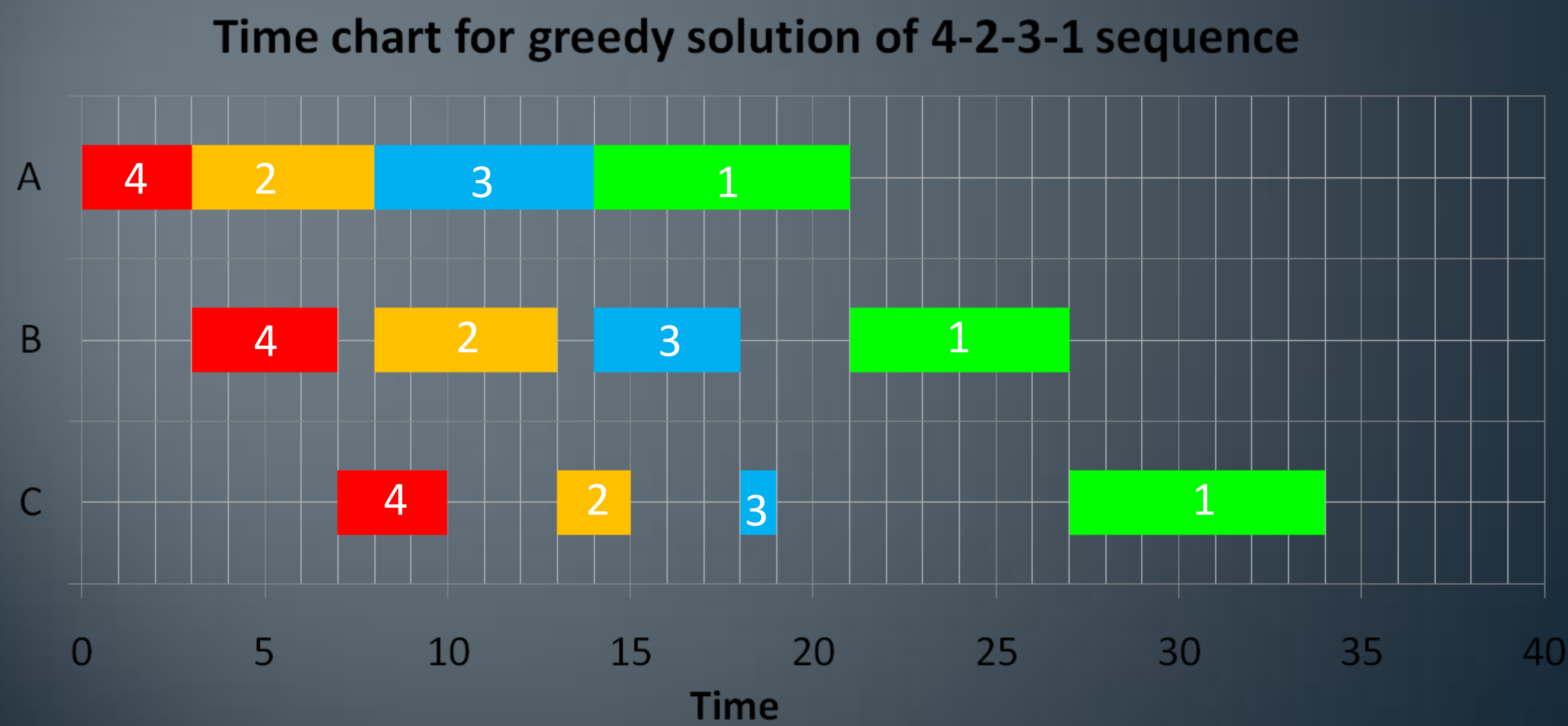
Greedy Algorithm

A possible greedy algorithm might start with selecting the fastest tasks for processor A.

Greedy solution

$T(4,2,3,1) = 34$

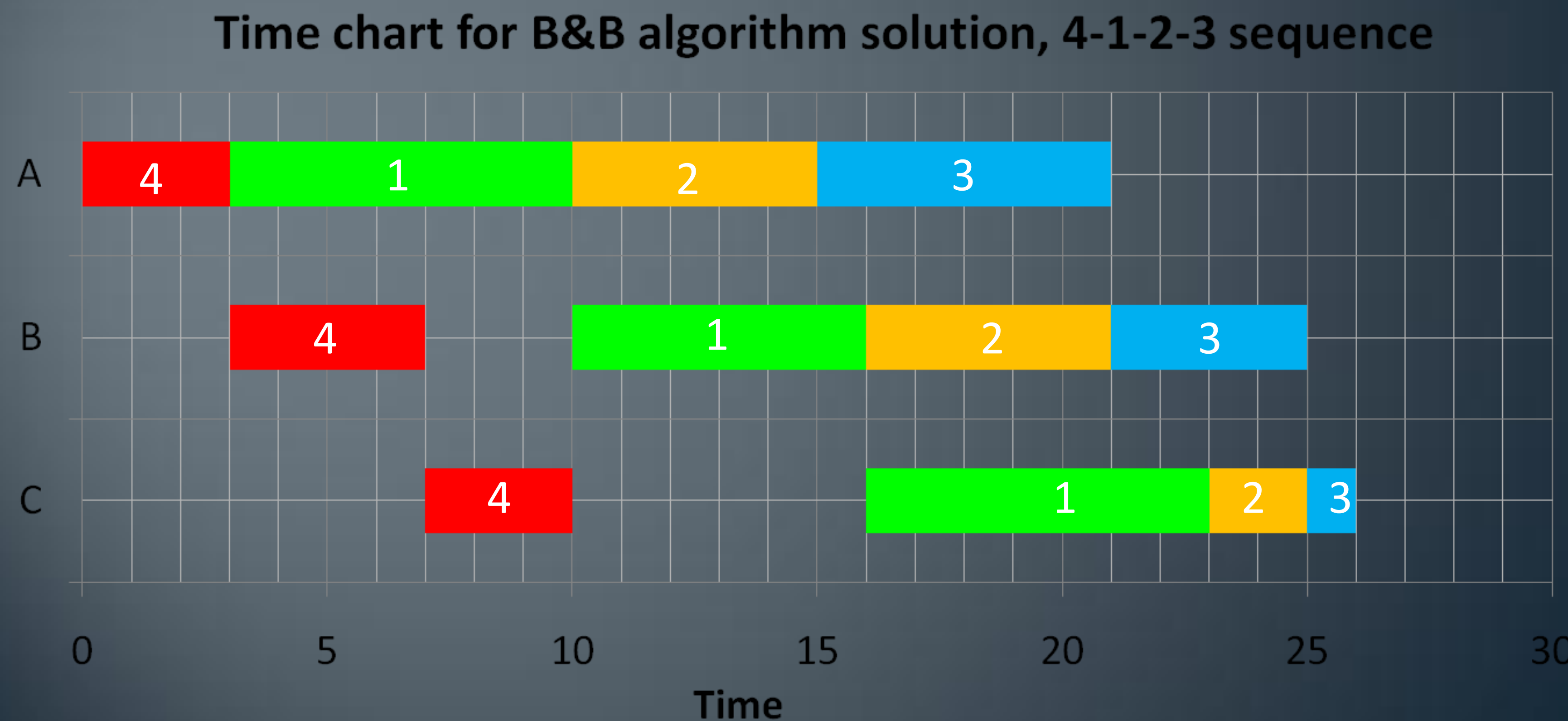
7	6	7
5	5	2
6	4	1
3	4	3



Optimal solution

$T(4,1,2,3) = 26$

7	6	7
5	5	2
6	4	1
3	4	3



Branch and bound Algorithm

Define a bounding criteria for a minimum time required by each branch of the decision tree

For level 1:

$$b(i) = A_i + \sum_{j=1}^{4} B_j + min_{j \neq i} C_j$$

For level 2:

$$b(i,j) = A_i + A_j + \sum_{k \neq i, k=1}^{4} B_k + min_{k \neq i,j} C_k$$

Level 1

$$b(i) = A_i + \sum_{j=1}^{4} B_j + min_{j \neq i} C_j$$

7	6	7
5	5	2
6	4	1
3	4	3

b(1)= 7+(6+5+4+4)+1=27

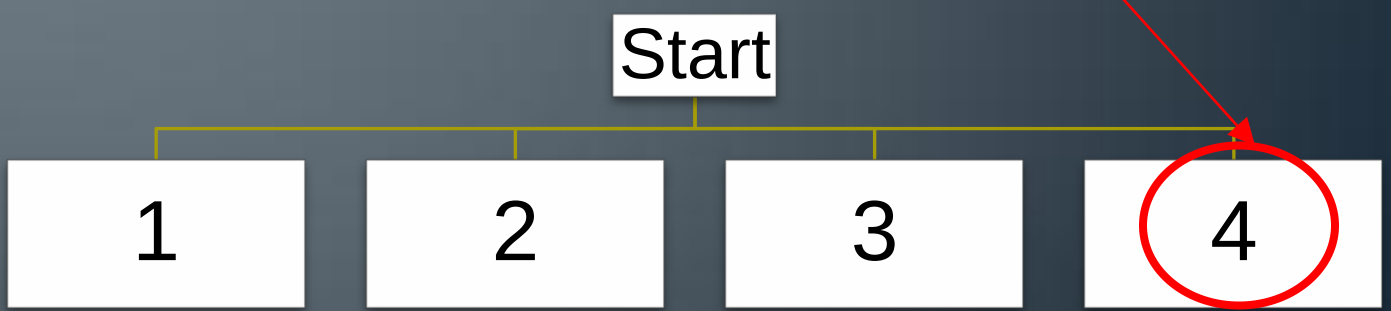
b(2)= 5+(6+5+4+4)+1=25

b(3)= 6+(6+5+4+4)+2=27

b(4)= 3+(6+5+4+4)+1=<u>23</u>

Minimum

This next



Bounds: T ≥27 T ≥25 T ≥27 T ≥23

Level 2

$$b(i,j) = A_i + A_j + \sum_{k \neq i, k=1}^{4} B_k + min_{k \neq i,j}\, C_k$$

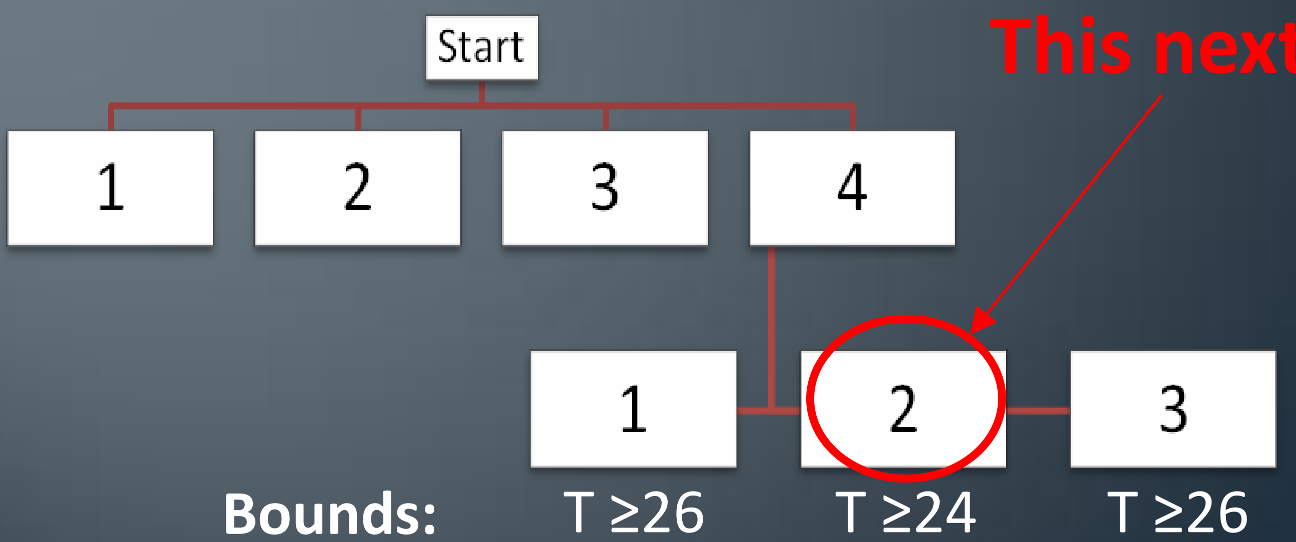
7	6	7
5	5	2
6	4	1
3	4	3

b(4,1)= (3+7)+(6+5+4)+1=26

b(4,2)= (3+5)+(6+5+4)+1=24

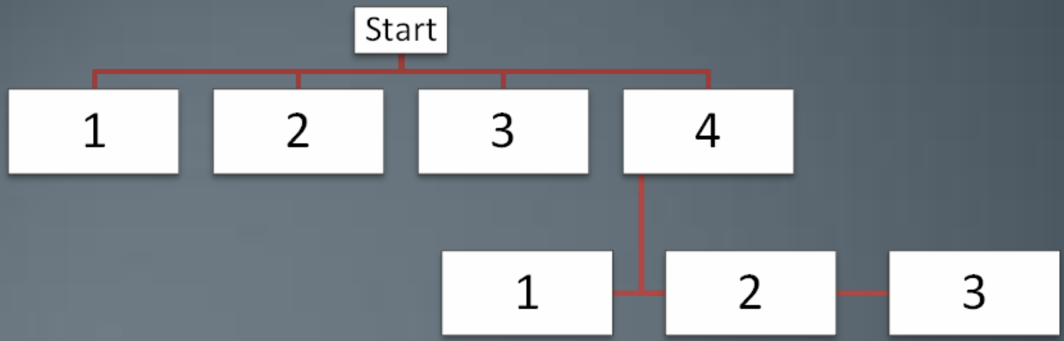
b(4,3)= (3+6)+(6+5+4)+2=26

Minimum



Solve the branch 4-2-x-x

7	6	7
5	5	2
6	4	1
3	4	3



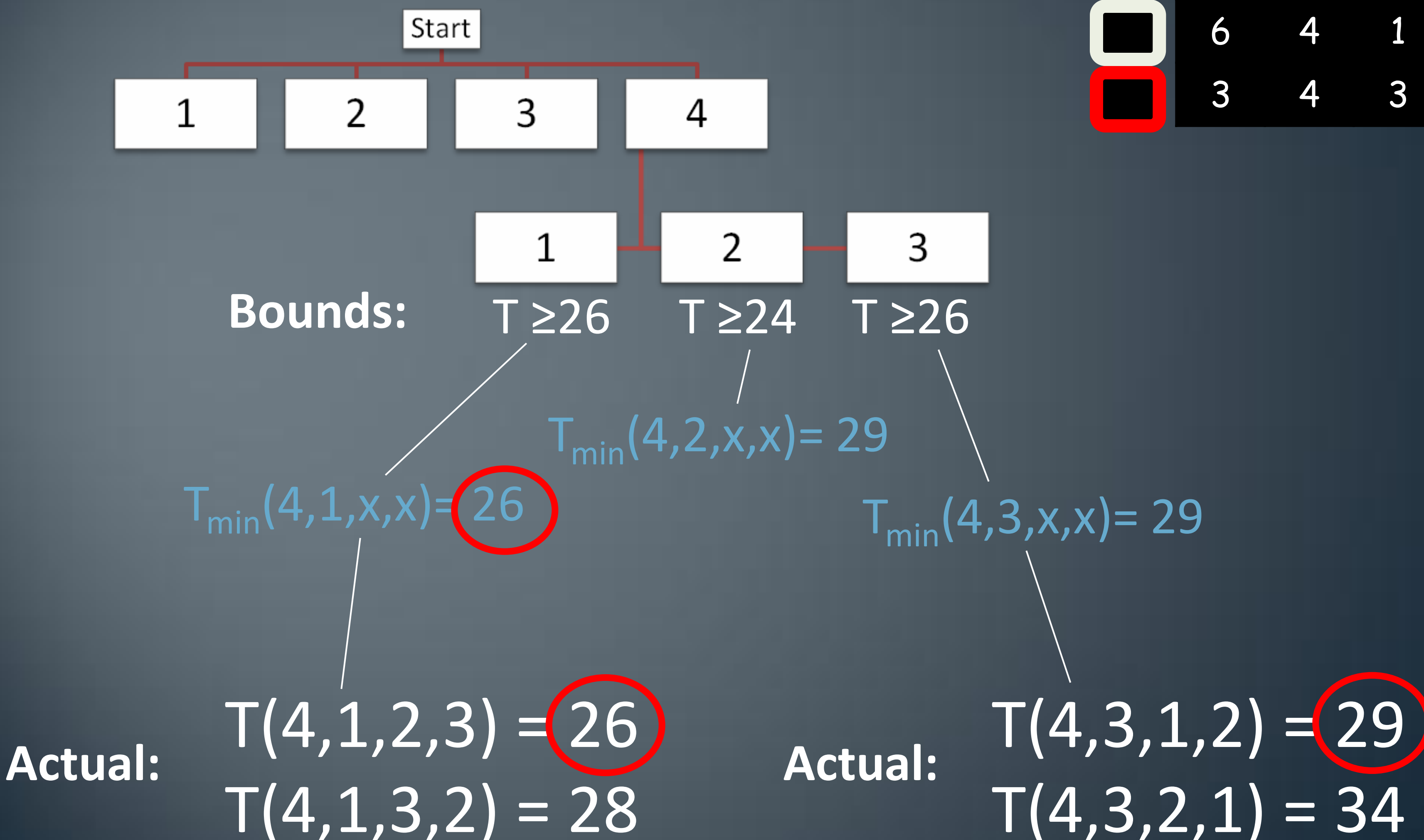
Bounds: $T \geq 26$ $T \geq 24$ $T \geq 26$

$T_{min}(4,2,x,x) = 29$

Actual:
$T(4,2,1,3) = 29$
$T(4,2,3,1) = 34$

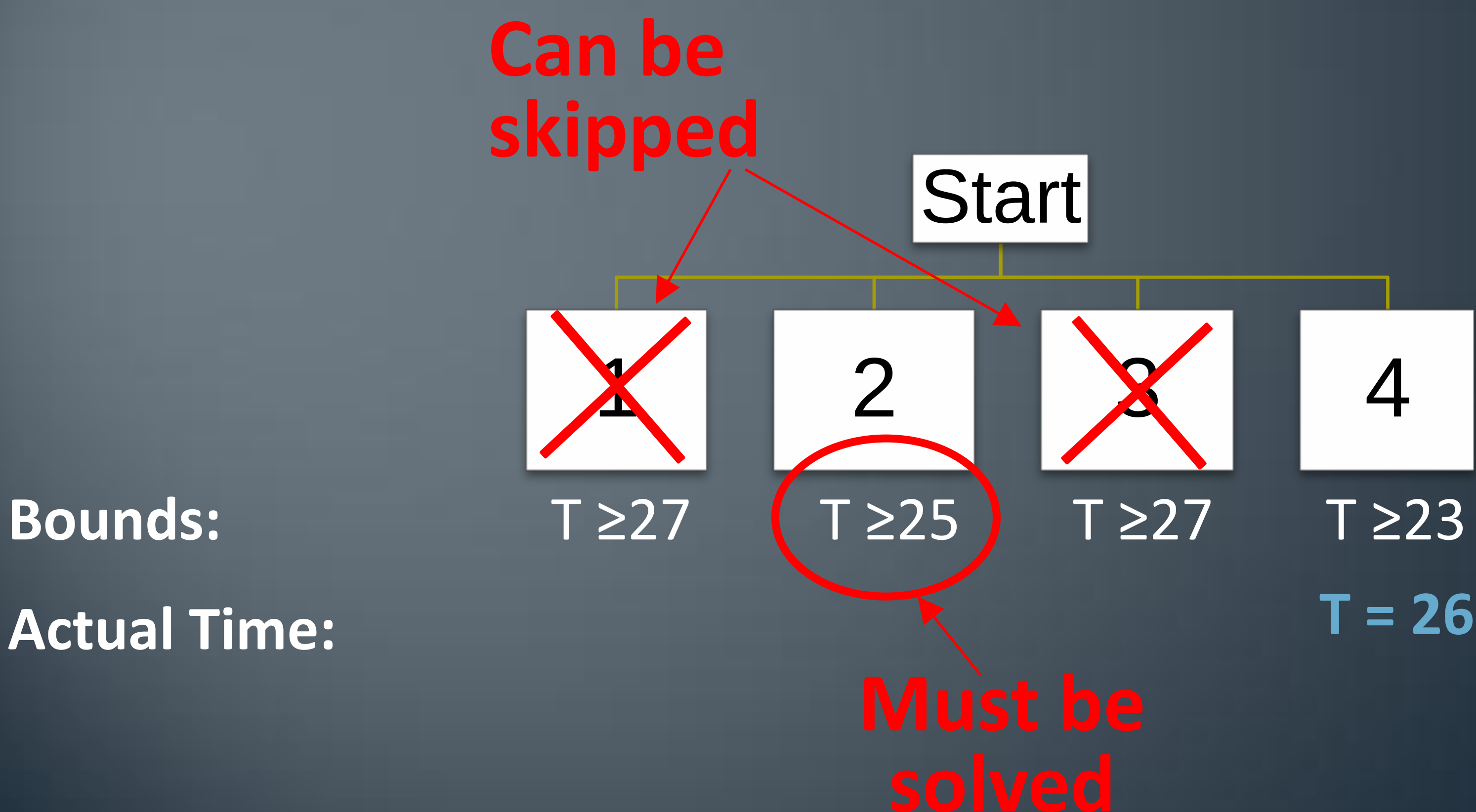
Solve the branch 4-2-x-x

7	6	7
5	5	2
6	4	1
3	4	3



Solve the other branches

7	6	7
5	5	2
6	4	1
3	4	3



$b(2,1) = (5+7)+(6+4+4)+1=27$

$b(2,3) = (5+6)+(6+4+4)+3=28$

$b(2,4) = (5+3)+(6+4+4)+1=\underline{23}$

The only candidate that can outperform T(4,1,2,3) is T(2,4,...) so we calculate it:

Actual T(2,4,1,3) = 29

Actual T(2,4,3,1) = 34

So the best time is T(4,1,2,3) and we don't need to solve the problem for any other branch because we now their minimum time, already.

7	6	7
5	5	2
6	4	1
3	4	3

Start

	1	2	3	4
Bounds:	$T \geq 27$	$T \geq 25$	$T \geq 27$	$T \geq 23$
Actual Time:		$T = 29$		**<u>T = 26</u>**

Bounds greater than 26!

Summary

- Using only the first level criteria we reduce the problem by 50% (omitting 2 main branches).
- Using the second level criteria we can reduce even more.