Tutorial 5

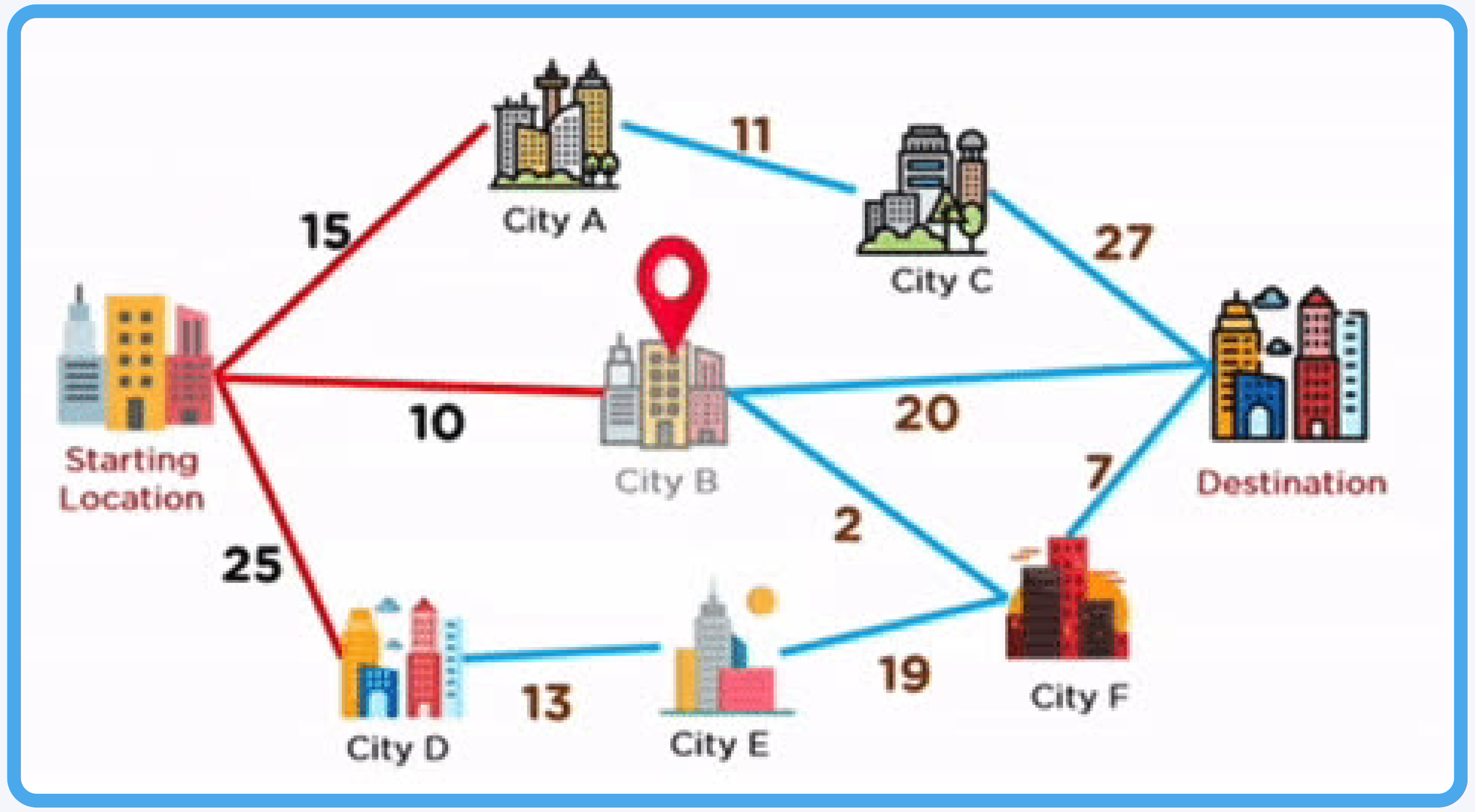
Analysis and Design of Algorithms

Greedy Algorithms

Greedy Algorithms

- **Greedy Algorithms** are a **programming approach** where we try to solve the problem by choosing **the best current option available.**

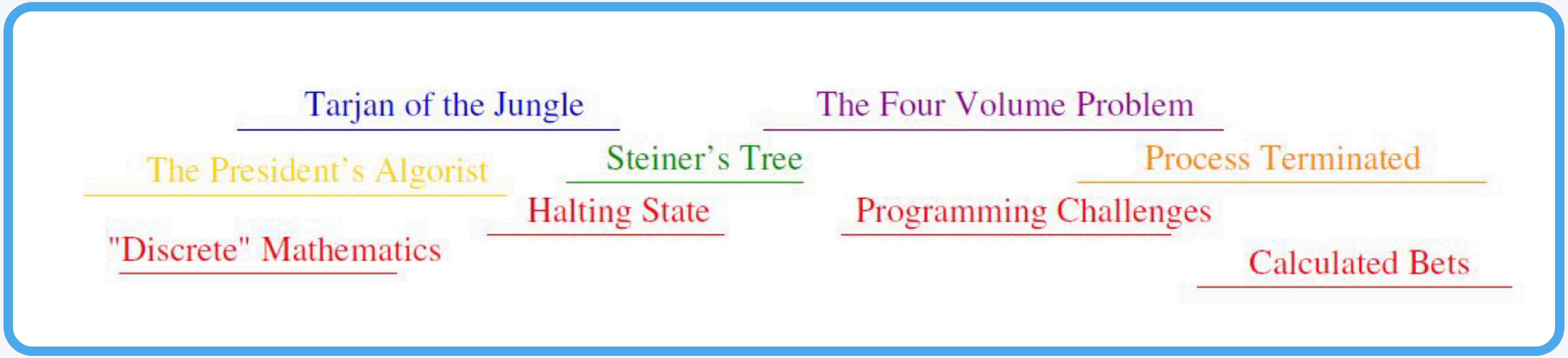
- This means we do not worry about whether **this choice is the overall best choice on the long run,** just that currently its **the best we've got.**



Important Terminology

- **Heuristic:** We choose the best local choice that we hope results in the optimal global solution, in order to guarantee speed.
- **Greedy Choice Property:** A problem with this property is a problem which can be solved **optimally** using the greedy approach.

Activity Scheduling



Tarjan of the Jungle

The Four Volume Problem

The President's Algorist

Steiner's Tree

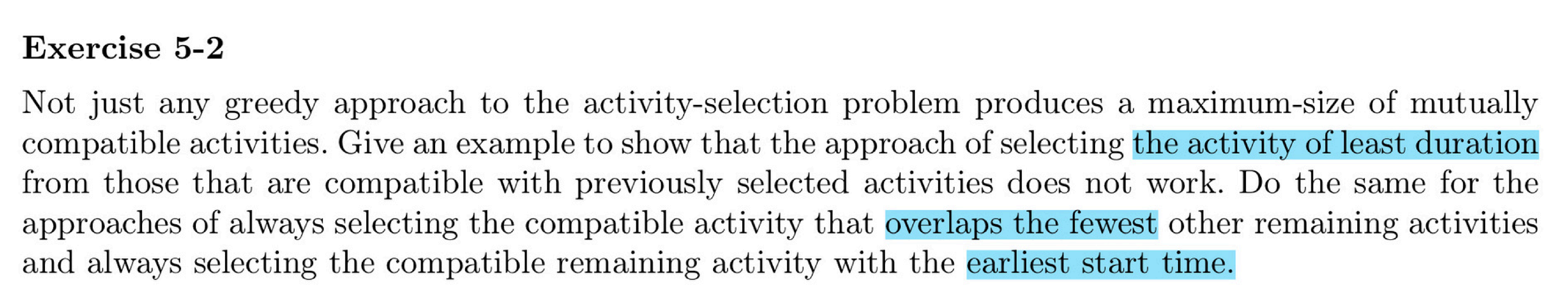
Process Terminated

Halting State

Programming Challenges

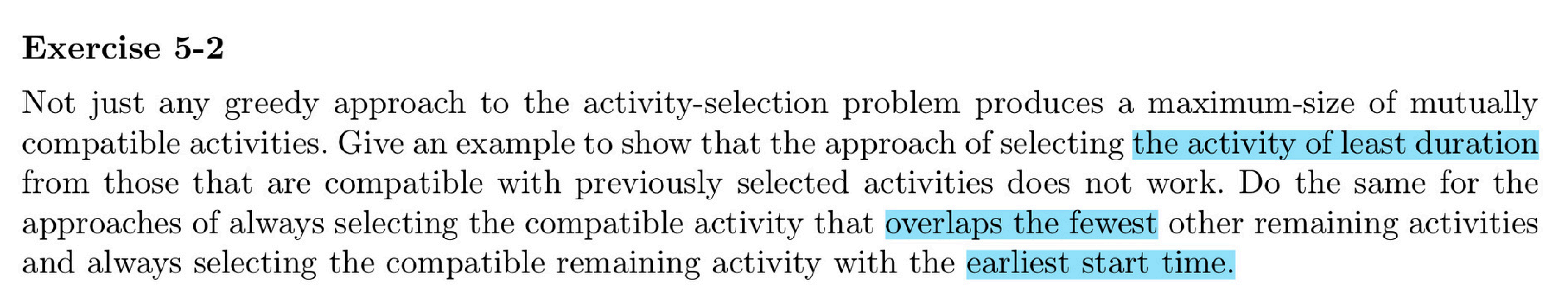
"Discrete" Mathematics

Calculated Bets



Exercise 5-2

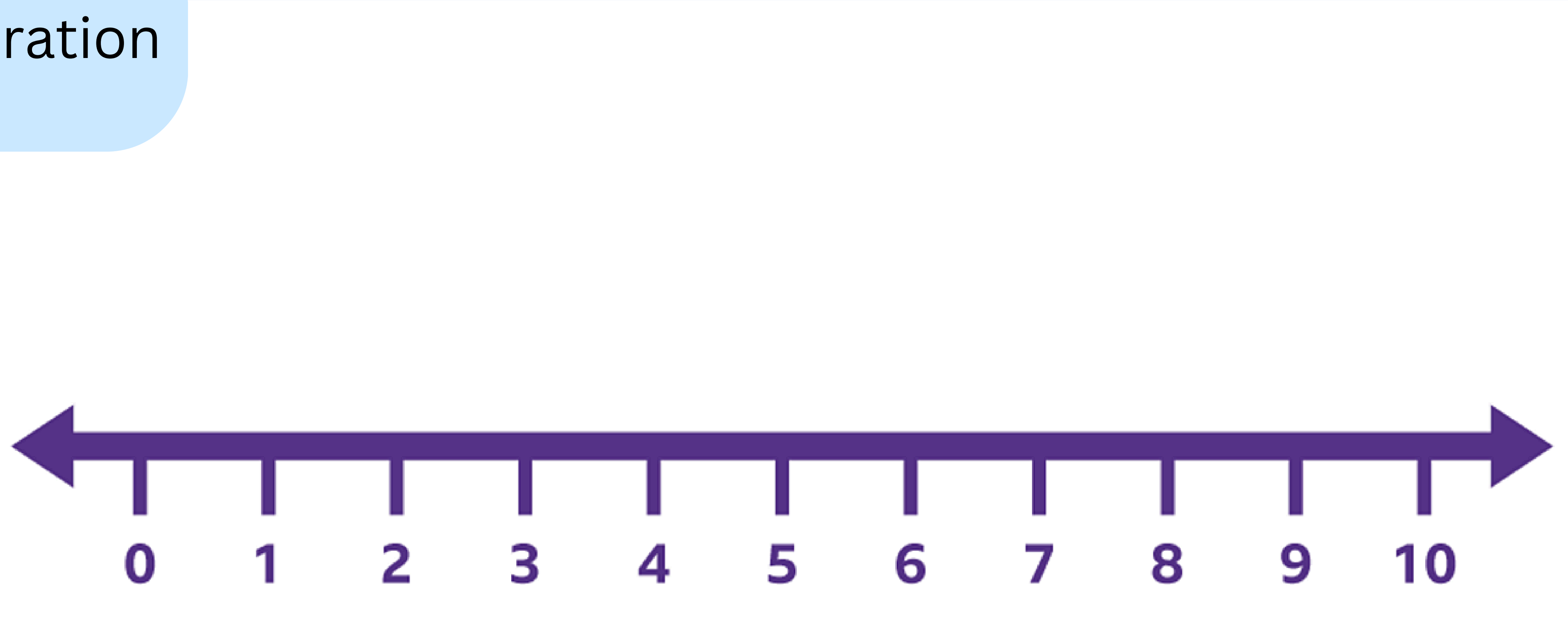
Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

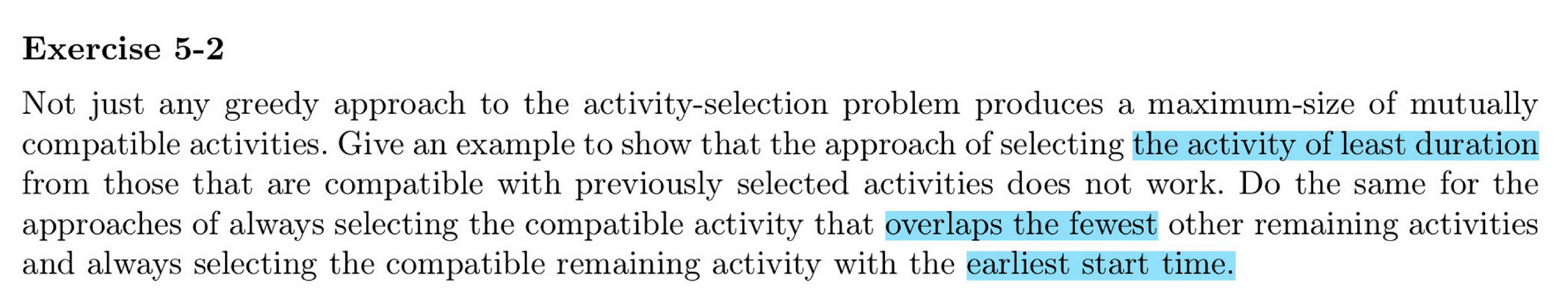


Exercise 5-2

Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

Least Duration

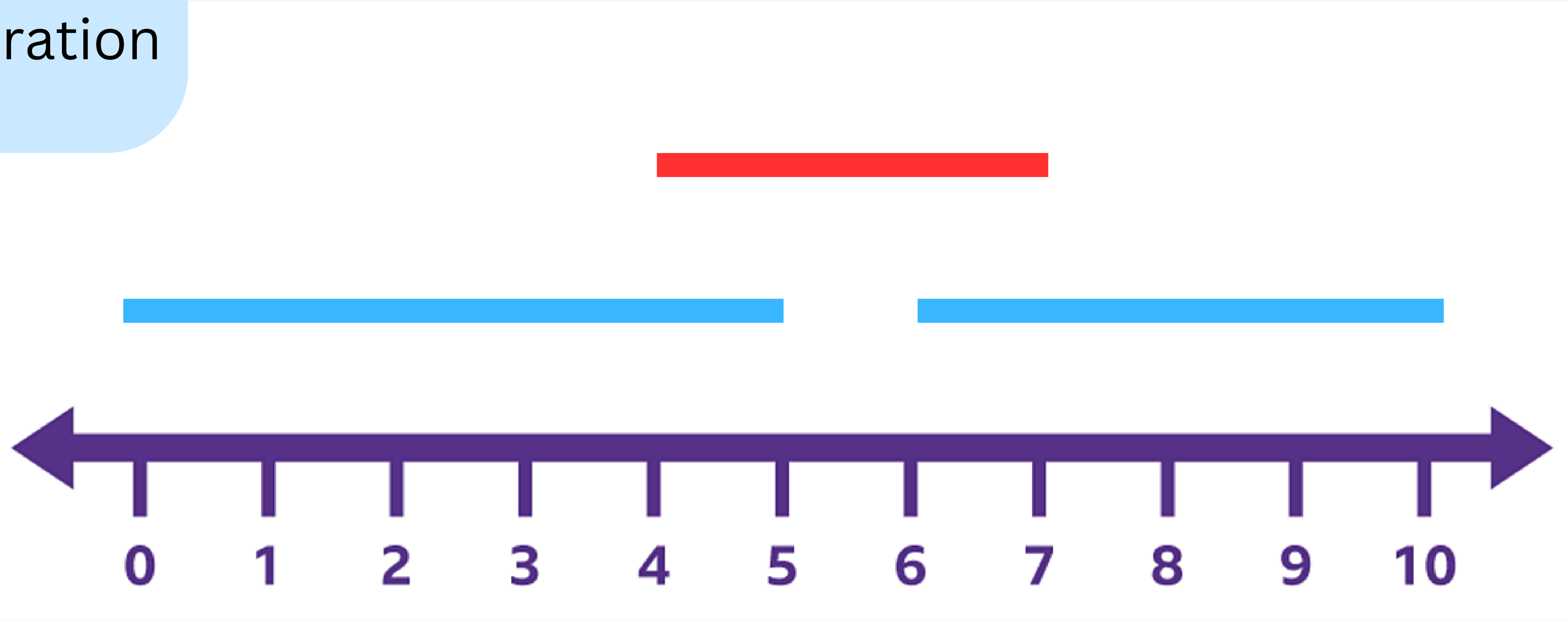


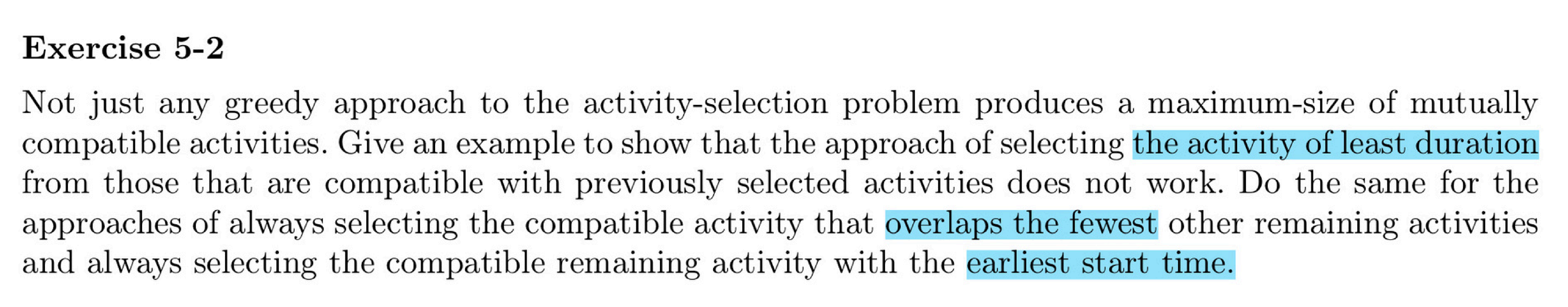


Exercise 5-2

Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

Least Duration

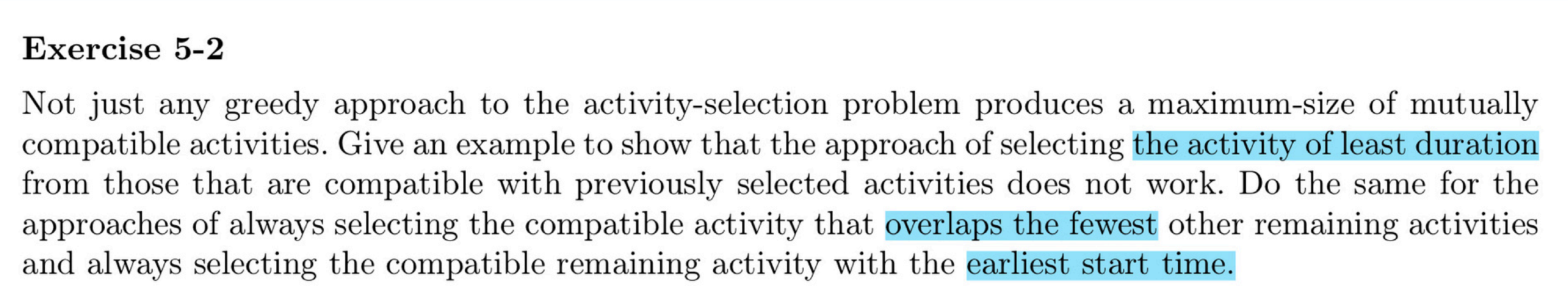




Exercise 5-2

Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

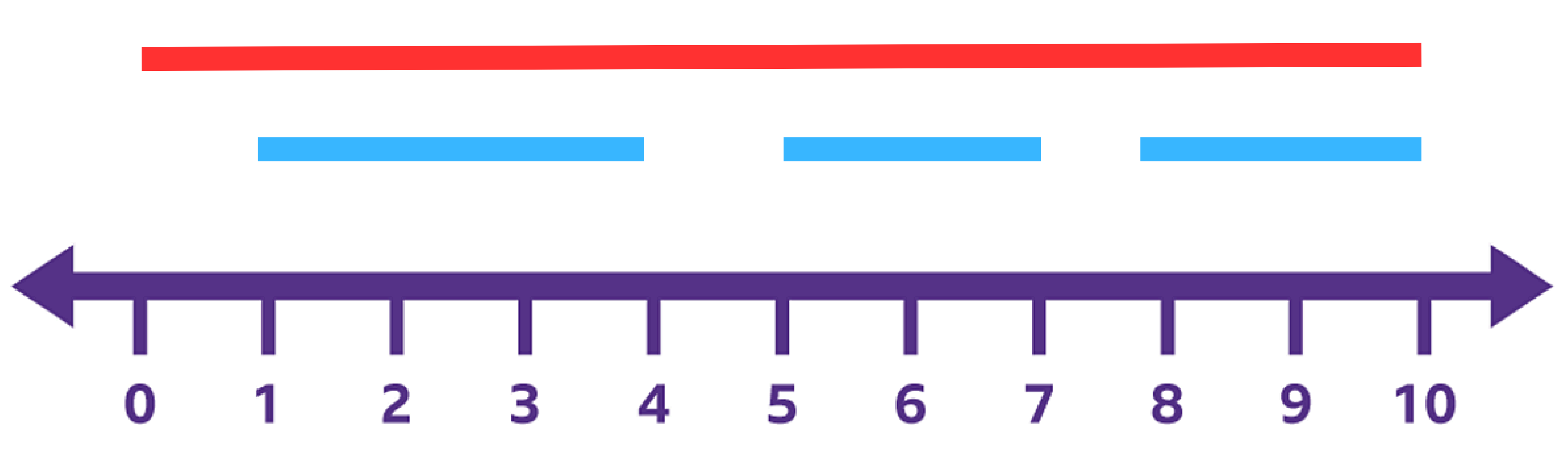
Earliest Start

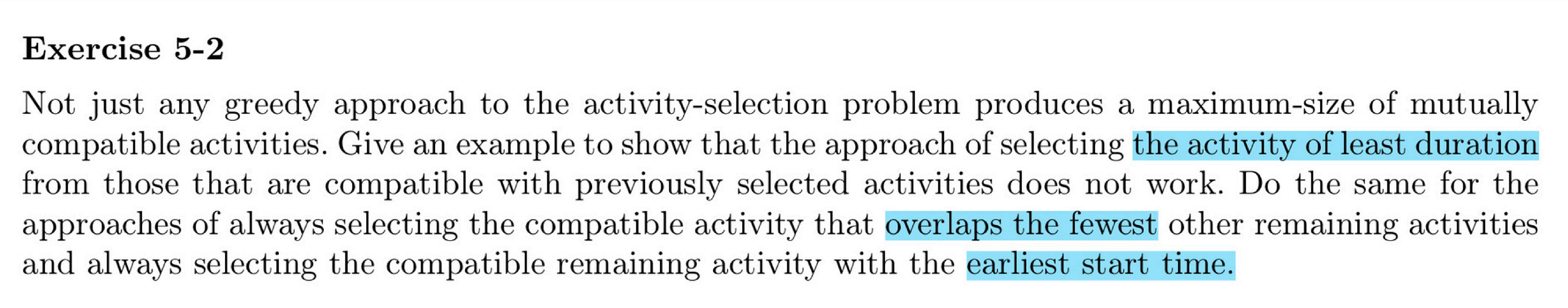


Exercise 5-2

Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

Earliest Start

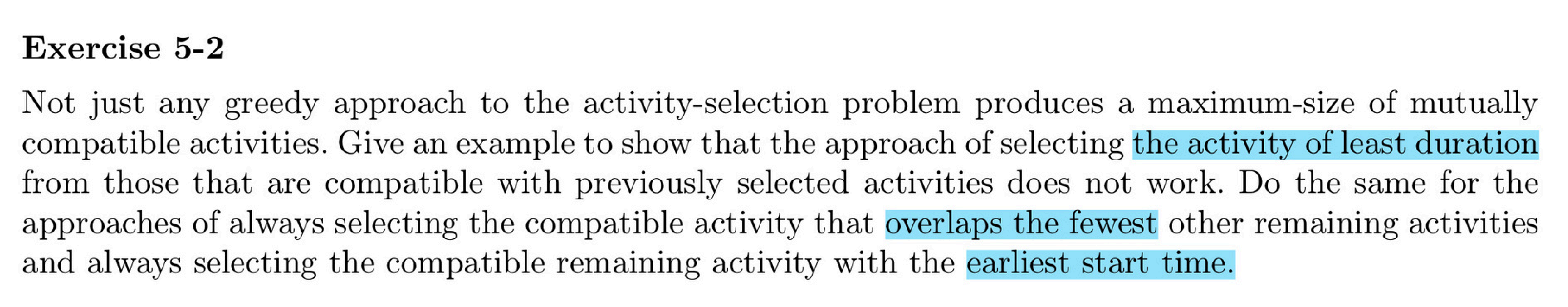




Exercise 5-2

Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

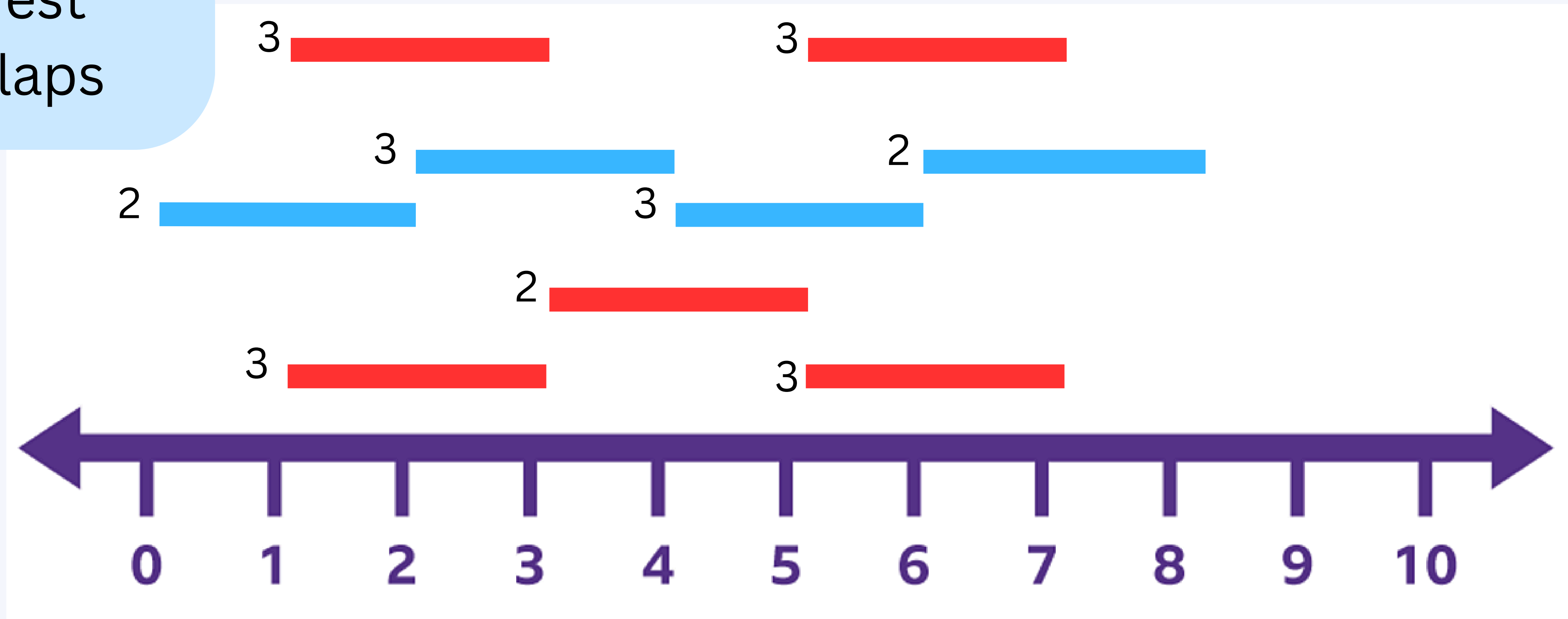
Fewest Overlaps



Exercise 5-2

Not just any greedy approach to the activity-selection problem produces a maximum-size of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

Fewest Overlaps



Exercise 5-3

Suppose a dentist has several patients waiting to be treated. Every patient i needs time t_i for treatment. The total time spent by all patients both waiting and being treated is referred to as the time in the system. A reasonable goal would be to minimize the time in the system.

i. Provide a greedy algorithm to reach this goal.

ii. Prove that your algorithm is optimal.

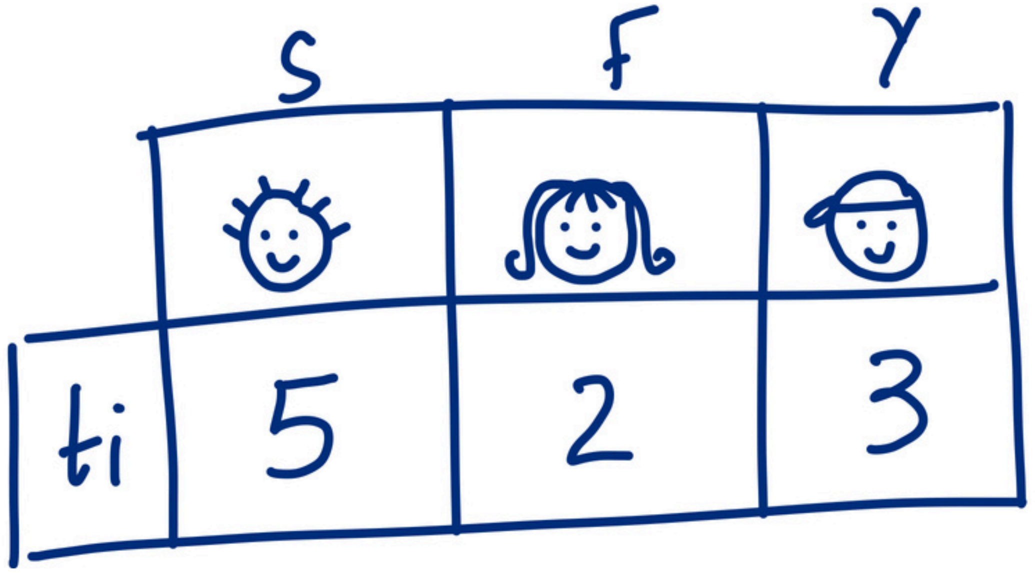
	S	F	Y
ti	5	2	3

Exercise 5-3

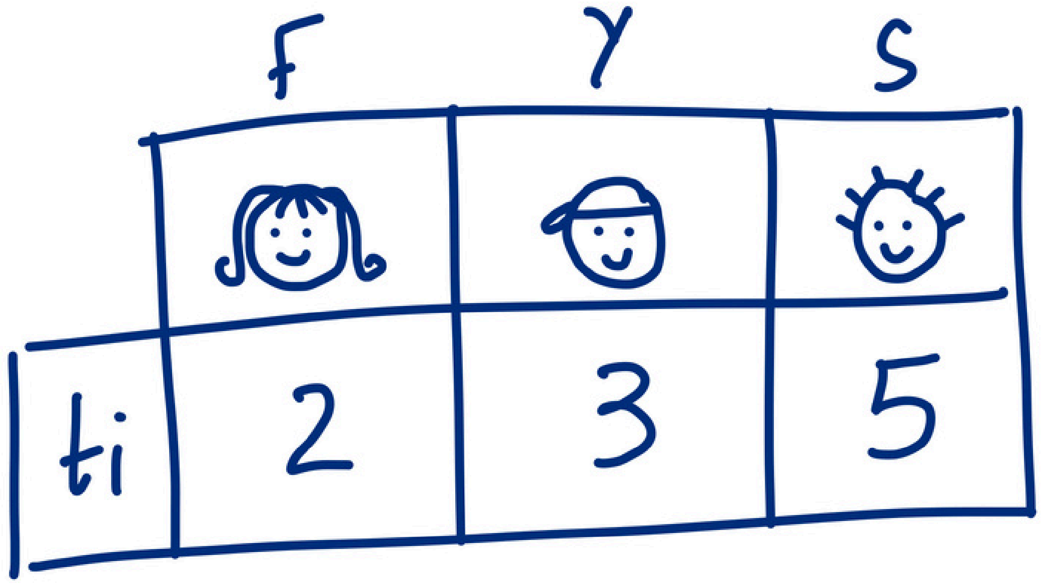
Suppose a dentist has several patients waiting to be treated. Every patient i needs time t_i for treatment. The total time spent by all patients both waiting and being treated is referred to as the time in the system. A reasonable goal would be to minimize the time in the system.

i. Provide a greedy algorithm to reach this goal.

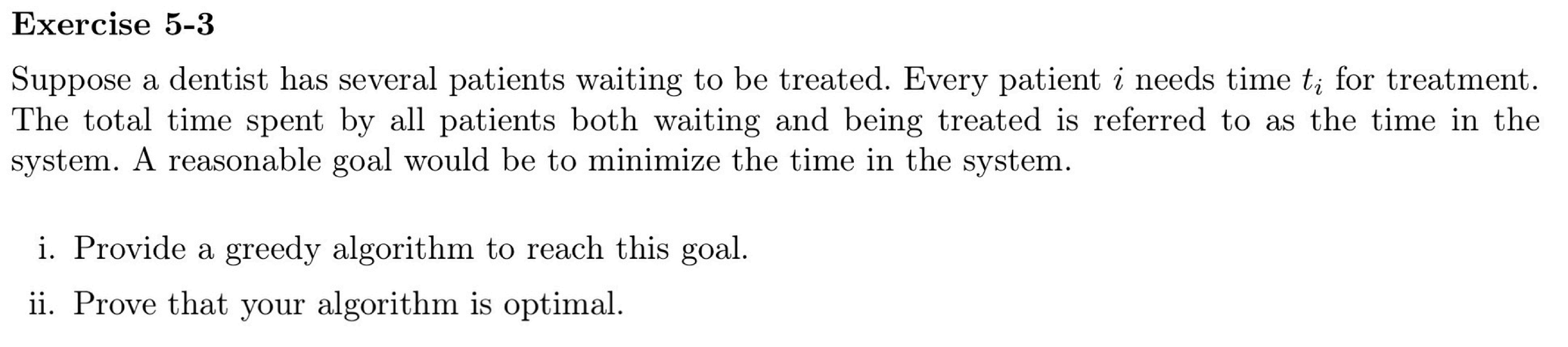
ii. Prove that your algorithm is optimal.



	s	f	y
t_i	5	2	3



	f	y	s
t_i	2	3	5

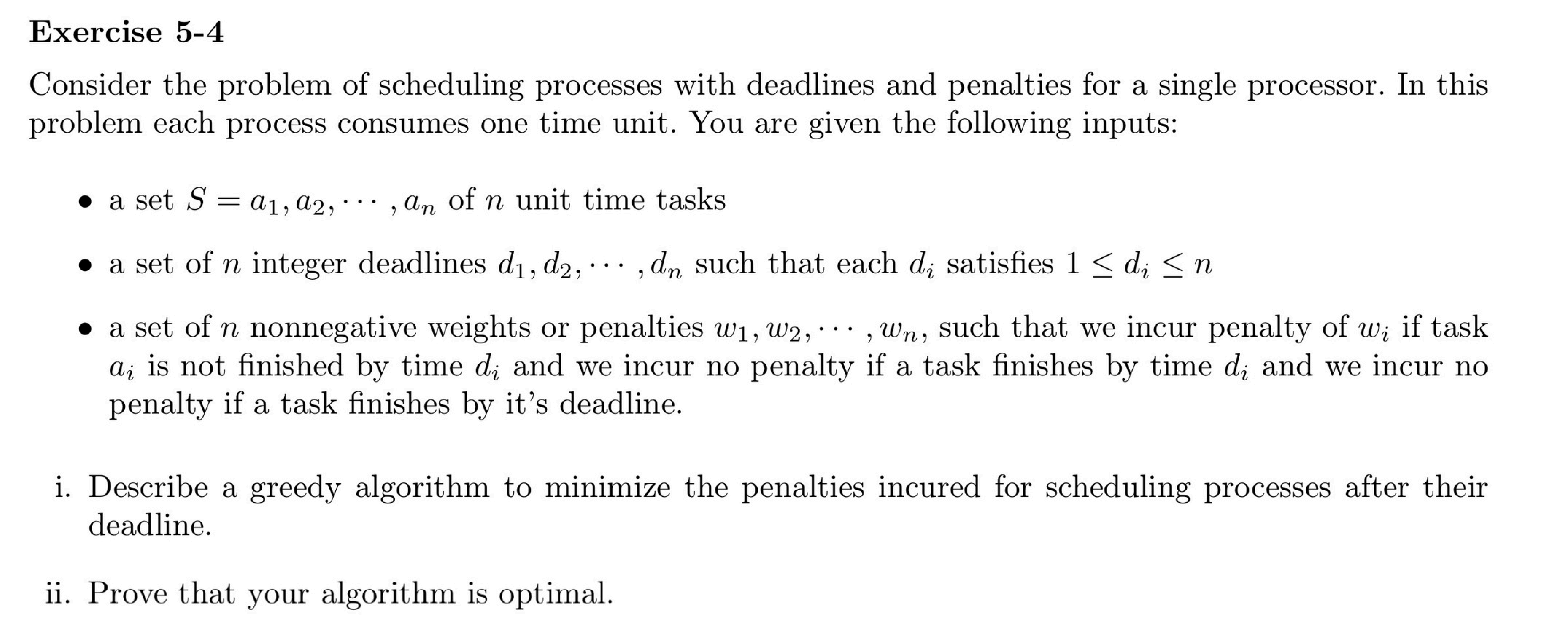


Exercise 5-3

Suppose a dentist has several patients waiting to be treated. Every patient i needs time t_i for treatment. The total time spent by all patients both waiting and being treated is referred to as the time in the system. A reasonable goal would be to minimize the time in the system.

i. Provide a greedy algorithm to reach this goal.

ii. Prove that your algorithm is optimal.



Exercise 5-4

Consider the problem of scheduling processes with deadlines and penalties for a single processor. In this problem each process consumes one time unit. You are given the following inputs:

- a set $S = a_1, a_2, \cdots, a_n$ of n unit time tasks
- a set of n integer deadlines $d_1, d_2, \cdots, d_n$ such that each d_i satisfies $1 \leq d_i \leq n$
- a set of n nonnegative weights or penalties $w_1, w_2, \cdots, w_n$, such that we incur penalty of w_i if task a_i is not finished by time d_i and we incur no penalty if a task finishes by time d_i and we incur no penalty if a task finishes by it's deadline.

i. Describe a greedy algorithm to minimize the penalties incured for scheduling processes after their deadline.

ii. Prove that your algorithm is optimal.

Example:

Original Input

S	*v*	*w*	*x*	*y*	*z*
D	*4*	*2*	*1*	*2*	*7*
W	*2*	*3*	*0*	*5*	*8*

Weight Desc.

S					
D					
W					

Deadline Asc.

S					
D					
W					

0
1
2
3
4
5
6
7
8
9
10

0
1
2
3
4
5
6
7
8
9
10

All done!