Chapter 3
Planning and Scheduling

Chapter Objectives

Check off these skills when you feel that you have mastered them.

- ☐ State the assumptions for the scheduling model.
- ☐ Compute the lower bound on the completion time for a list of independent tasks on a given number of processors.
- ☐ Describe the list-processing algorithm.
- ☐ Apply the list-processing algorithm to schedule independent tasks on identical processors.
- ☐ For a given list of independent tasks, compare the total task time using the list-processing algorithm for both the non-sorted list and also a decreasing-time list.
- ☐ When given an order-requirement digraph, apply the list-processing algorithm to schedule a list of tasks subject to the digraph.
- ☐ Explain how a bin-packing problem differs from a scheduling problem.
- ☐ Given an application, determine whether its solution is found by the list-processing algorithm or by one of the bin-packing algorithms.
- ☐ Discuss advantages and disadvantages of the next-fit bin-packing algorithm.
- ☐ Solve a bin-packing problem by the non-sorted next-fit algorithm.
- ☐ Solve a bin-packing problem by the decreasing-time next-fit algorithm.
- ☐ Discuss advantages and disadvantages of the first-fit bin-packing algorithm.
- ☐ Apply the non-sorted first-fit algorithm to a bin-packing problem.
- ☐ Apply the decreasing-time first-fit algorithm to a bin-packing problem.
- ☐ Discuss advantages and disadvantages of the worst-fit bin-packing algorithm.
- ☐ Find the solution to a bin-packing problem by the non-sorted best-fit algorithm.
- ☐ Find the solution to a bin-packing problem by the decreasing-time worst-fit algorithm.
- ☐ List two examples of bin-packing problems.
- ☐ Create a vertex coloring of a graph, and explain its meaning in terms of assigned resources.
- ☐ Find the chromatic number of a graph.
- ☐ Interpret a problem of allocation of resources with conflict as a graph, and find an efficient coloring of the graph.

Guided Reading

Introduction
We consider efficient ways to schedule a number of related tasks, with constraints on the number of workers, machines, space, or time available.

Section 3.1 Scheduling Tasks

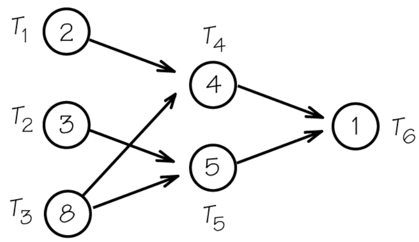
Key idea
The **machine-scheduling problem** is to decide how a collection of tasks can be handled by a certain number of **processors** as quickly as possible. We have to respect both order requirements among the tasks and a priority list.

Key idea
The **list-processing algorithm** chooses a task for an available processor by running through the priority list in order, until it finds the **ready task**.

Example A
In this digraph, which tasks are ready

a) as you begin scheduling?

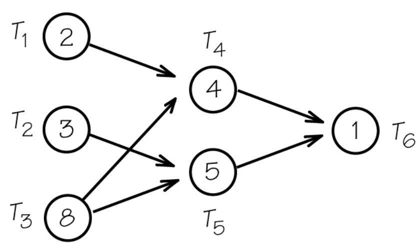
b) if just T_2 and T_3 have been completed?



Solution
a) At the start, only T_1, T_2, and T_3 have no required predecessors.

b) With T_2 and T_3 completed, T_5 is ready but T_4 must wait for T_1. Also, T_6 must wait for T_4 and T_5.

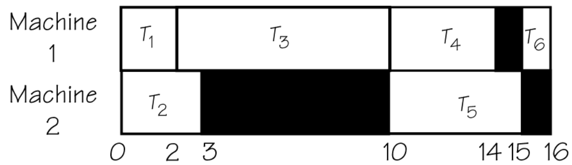
Example B
Schedule the tasks in the digraph on two processors with priority list $T_1, T_2, T_3, T_4, T_5, T_6,$ and determine the completion time.



Continued on next page

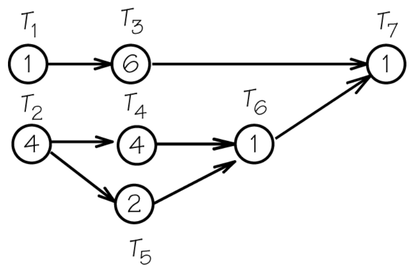
Solution

T_1 and T_2 are first, then T_3. T_4 and T_5 must wait for completion of T_3. Also, T_6 must wait for T_4 and T_5. The completion time is 16. The following is the schedule.



Question 1

Schedule the tasks in the digraph on two processors with priority list T_1, T_2, T_3, T_4, T_5, T_6, T_7. What is the completion time?



Answer

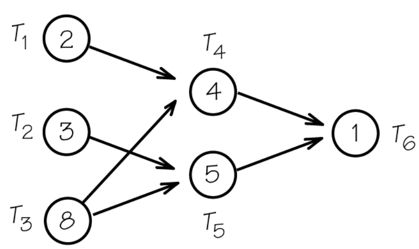
The completion time is 11.

Key idea

Different priority lists can lead to different schedules and completion times.

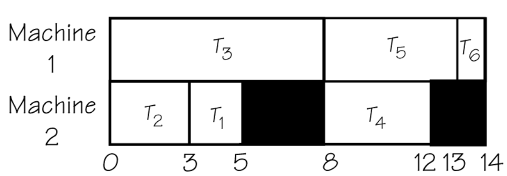
Example C

Schedule the tasks in the digraph on two processors with priority list T_6, T_5, T_4, T_3, T_2, T_1, and determine the completion time.



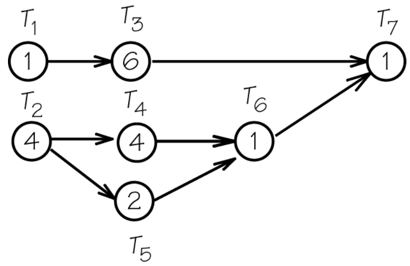
Solution

T_3 is the highest priority ready task and gets scheduled first, with T_2 next. After T_2 is done, no task is ready except for T_1, so T_1 is scheduled. When T_3 is done, T_6 is not ready so T_5 is scheduled, followed by T_4 and T_6. The completion time is 14. The following is the schedule:



Question 2

Schedule the tasks in the digraph on two processors with priority list $T_7, T_6, T_5, T_4, T_3, T_2, T_1$. What is the completion time?



Answer

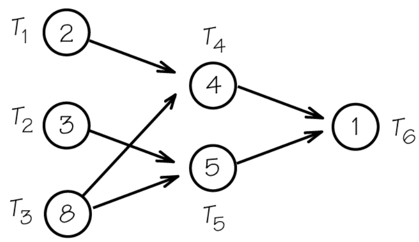
The completion time is 12.

Key idea

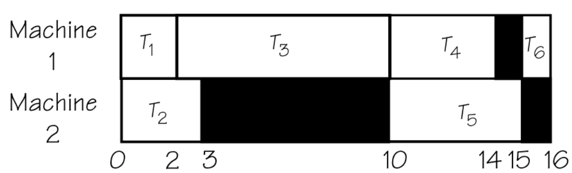
A schedule is optimal if it has the earliest possible total completion time. For example, a critical path in the order-requirement digraph may determine the earliest completion time.

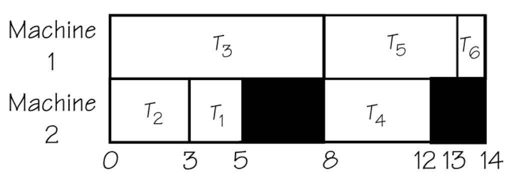
Example D

Find a critical path in the digraph.



Is one of the schedules constructed optimal? If so, which one?



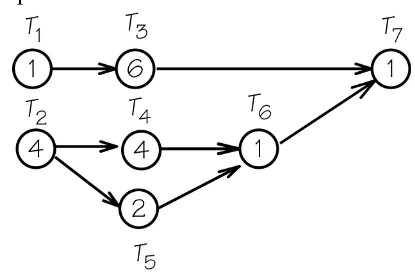


Solution

T_3, T_5, T_6 is the longest path in the digraph, and no scheduling can be completed in less time than the length of this path. Because the second schedule is completed at the same time of 14, it matches the critical path length of 14. Thus, the second schedule is optimal.

Question 3

What is the length of the critical path?



Answer

10

Section 3.2 Critical-Path Schedules

Key idea

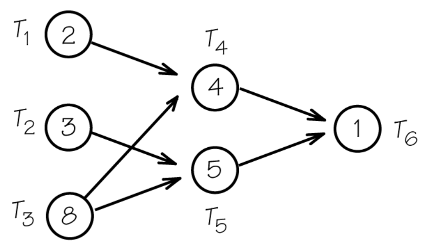
If we can choose or change a priority list, then we have a chance to find an optimal schedule.

Key idea

The **critical-path scheduling** algorithm schedules first the tasks in a critical path.

Example E

Use critical-path scheduling to construct a priority list for the tasks in this digraph.



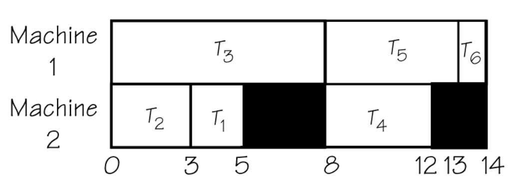
Solution

T_3 is at the head of a critical path. When you remove T_3 and its arrows, T_2 is the head of the remaining critical path T_2, T_5, T_6. Removing T_2 and its arrows makes T_1 the head of T_1, T_4, T_6. Finally, T_5 is the head of T_5, T_6. Thus the priority list is T_3, T_2, T_1, T_5, T_4, T_6.

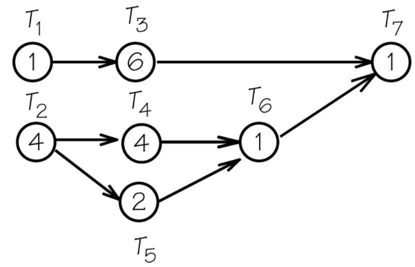
Example F

Construct the schedule for the tasks on two processors based on the critical path priority list from the above, T_3, T_2, T_1, T_5, T_4, T_6.

Solution



Question 4
Use critical-path scheduling to construct a priority list for the tasks in this digraph. How much total time are the two machines not processing tasks?



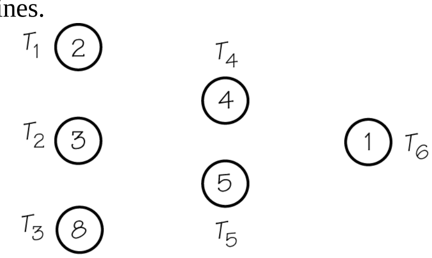
Answer
3

Section 3.3 Independent Tasks

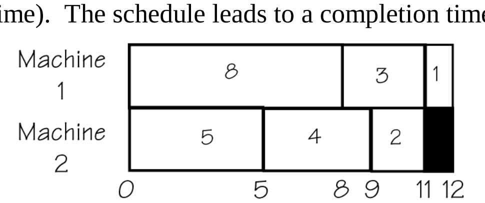
Key idea
When a set of tasks are **independent** (can be done in any order), we have a variety of available algorithms to choose a priority list leading to close-to-optimal scheduling. Some algorithms perform well in the **average-case**, but poorly in the **worst-case**.

Key idea
The **decreasing-time-list algorithm** schedules the longest tasks earliest. By erasing the arrows in the digraph used throughout this chapter, we obtain a set of independent tasks.

Example G
Construct a decreasing-time priority list. Use this list to schedule the tasks and determine the completion time on two machines.

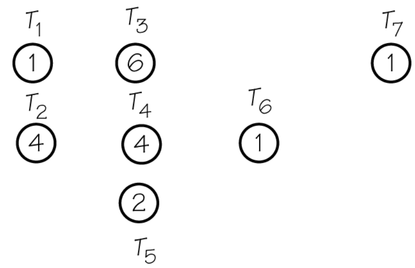


Solution
The list is T_3, T_5, T_4, T_2, T_1, T_6. Since we treat these as independent tasks, the task reference can be removed (keeping only the time). The schedule leads to a completion time of 12.



Question 5

Construct a decreasing-time priority list. What is the completion time on two machines?



Answer

10

Section 3.4 Bin Packing

Key idea

With the **bin-packing problem**, we consider scheduling tasks within a fixed time limit, using as few processors as possible. This is like fitting boxes into bins of a certain size − but it is used in a variety of real-world applications.

Key idea

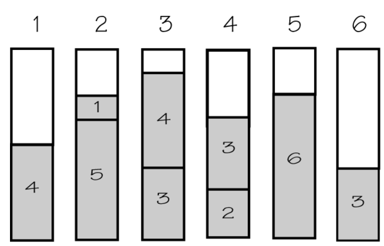
We have a variety of heuristic algorithms available to do the packing well if not optimally. Three important algorithms are **next fit (NF)**, **first fit (FF)**, and **worst fit (WF).**

Example H

Use the next-fit algorithm to pack boxes of sizes 4, 5, 1, 3, 4, 2, 3, 6, 3 into bins of capacity 8. How many bins are required?

Solution

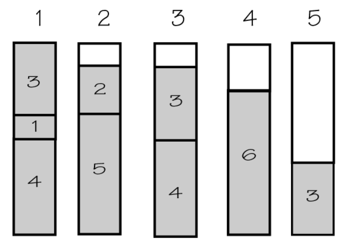
There are six bins required.



Bin 1 did not have enough space left for the second box, so bin 2 was used. There was enough room in bin 1 for the third box, but the NF heuristic doesn't permit us to go back to earlier bins. Once a bin is opened, it is used as long as the boxes fit − if they don't fit, a new bin is opened.

Example I
Now use the first-fit algorithm to pack boxes of sizes 4, 5, 1, 3, 4, 2, 3, 6, 3 into bins of capacity 8. How many bins are required?

Solution
There are five bins required.



After opening bin 2 for the second box, we were able to go back to bin 1 for the next two boxes. The FF heuristic allows us to return to earlier bins while the NF does not.

Key idea
In the worst-fit (WF) algorithm we pack an item into a bin with the most room available. Although this algorithm can lead to the same number of bins as other algorithms, the items may be packed in a different order.

Key idea
WF is like FF in that it permits returning to earlier bins. However, in FF you always start back at the first bin and sequentially search for a bin that will accommodate this weight, while in WF you calculate the unused space in each available bin and select the bin with the maximum room.

Question 6
Consider packing boxes sized 2, 6, 2, 6, 3, 4, 1, 4, 2 into bins of capacity 7. How many bins are required if we pack using

a) next-fit algorithm?

b) first-fit algorithm?

c) worst-fit algorithm?

Answer
a) 6

b) 5

c) 5

Key idea

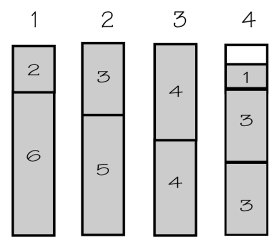
Each of these algorithms can be combined with **decreasing-time** heuristics, leading to the three algorithms **next-fit decreasing (NFD), first-fit decreasing (FFD),** and **worst-fit decreasing (WFD**).

Example J

Use the first-fit decreasing algorithm to pack the boxes of sizes 4, 5, 1, 3, 4, 2, 3, 6, 3 into bins of capacity 8. How many bins are required?

Solution

First, rearrange the boxes in size decreasing order: 6, 5, 4, 4, 3, 3, 3, 2, 1.



There are four bins required.

Example K

Is any of the algorithms you have used to pack the boxes sized 4, 5, 1, 3, 4, 2, 3, 6, 3 an optimal packing (that is, one using the fewest bins)?

Solution

In this case, FFD found the optimal packing. The amount of unused space is obviously less than the capacity of one bin. Therefore, no fewer than four bins could be used to hold all the boxes in this problem. However, neither FFD nor any of the other heuristics discussed in this section will necessarily find the optimal number of bins in an arbitrary problem.

Question 7

Consider packing boxes sized 2, 6, 2, 6, 3, 4, 1, 4, 2 into bins of capacity 7. How many bins are required if we pack using

a) next-fit decreasing algorithm?

b) first-fit decreasing algorithm?

Answer

a) 5

b) 5

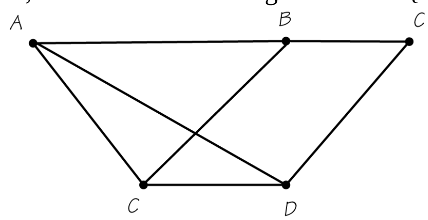
Section 3.5 Resolving Conflicts via Coloring

Key idea

If we represent items to be scheduled (classes, interviews, etc) as vertices in a graph, then a **vertex coloring** of the graph can be used to assign resources, such as times or rooms, to the items in a conflict-free manner.

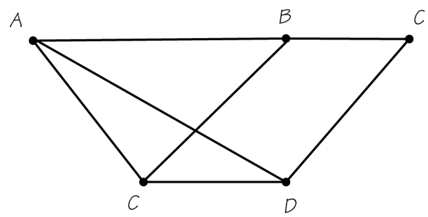
Key idea

The **chromatic number** of the graph determines the minimum amount of the resource that must be made available for a conflict-free schedule.
Here is a graph with five vertices, which is colored using four colors {*A*, *B*, *C*, *D*}.



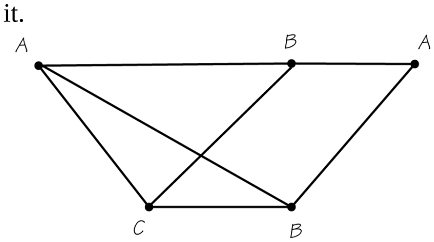
Example L

Discuss the coloring of the following graph. Can you find a vertex coloring of the same graph with only three colors?



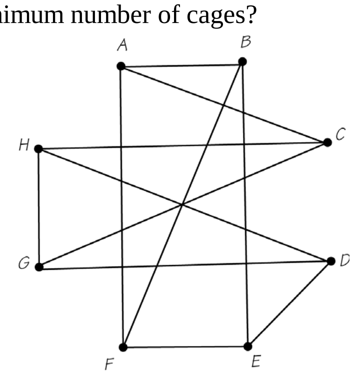
Solution

The vertex labeled *A*, for example, is connected to three other vertices, labeled *C*, *B*, *D*. No vertex is connected to four others. It is possible to find a vertex coloring of the same graph with only three colors. Here is one way to do it.



Question 8

Suppose the following graph shows conflicts between animals A – H. If an edge connects two animals then they cannot be put in the same cage. Determine a suitable arrangement with a minimum number of cages. What is the minimum number of cages?



Answer

3

Homework Help

Exercises 1 – 3, 8
Answers will vary. Think of these as real-world situations.

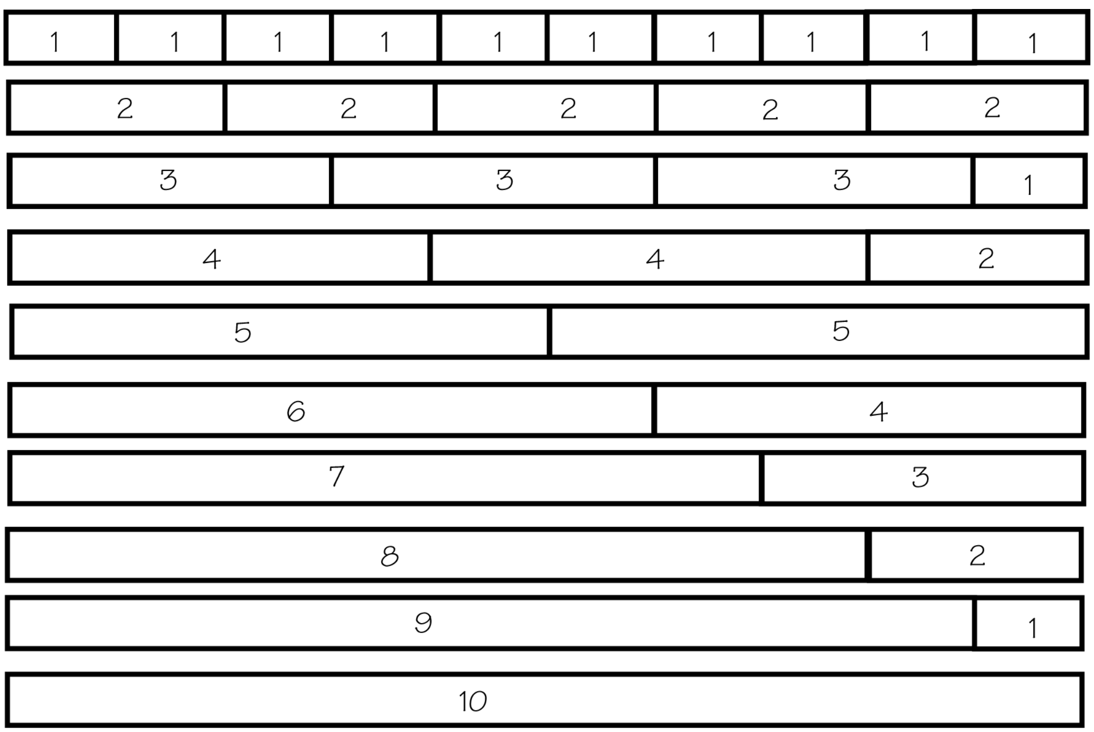
Exercises 4 – 7, 9 – 26
Carefully read through Section 3.1 and 3.2 and their examples. Recall the length of the critical path is the longest path.

Exercises 27 – 41
Carefully read through Section 3.3 and its examples. With independent tasks, you do not need to be concerned about tasks preceding the ones you are scheduling.

Exercises 42 – 62
Carefully read through Section 3.4 and its examples. Be careful when applying the different methods of packing. Although the same number of bins may be used, they may be packed differently depending on the method.

Exercises 63 – 78
Carefully read through Section 3.5 and its examples. It is possible that two colorings of a graph are correct as long as they use the minimum number of colors.

Below are blocks that can be cut out in order to help you with scheduling machines and bin packing.



Do You Know the Terms?

Cut out the following 20 flashcards to test yourself on Review Vocabulary. You can also find these flashcards at http://www.whfreeman.com/fapp7e.

Chapter 3 **Planning and Scheduling** **Average-case analysis**	**Chapter 3** **Planning and Scheduling** **Bin-packing problem**
Chapter 3 **Planning and Scheduling** **Chromatic number**	**Chapter 3** **Planning and Scheduling** **Critical-path scheduling**
Chapter 3 **Planning and Scheduling** **Decreasing-time-list algorithm**	**Chapter 3** **Planning and Scheduling** **First fit (FF)**
Chapter 3 **Planning and Scheduling** **First-fit decreasing (FFD)**	**Chapter 3** **Planning and Scheduling** **Heuristic algorithm**

The problem of determining the minimum number of containers of capacity W into which objects of size $w_1, ..., w_n \ (w_i \leq W)$ can be packed.	**The study of the list-processing algorithm (more generally, any algorithm) from the point of view of how well it performs in all the types of problems it may be used for and seeing on average how well it does. *See also* worst-case analysis.**
A heuristic algorithm for solving scheduling problems where the list-processing algorithm is applied to the priority list obtained by listing next in the priority list a task that heads a longest path in the order-requirement digraph. This task is then deleted from the order-requirement digraph, and the next task placed in the priority list is obtained by repeating the process.	**The chromatic number of a graph *G* is the minimum number of colors (labels) needed in any vertex coloring of *G*.**
A heuristic algorithm for bin packing in which the next weight to be packed is placed in the lowest-numbered bin already opened into which it will fit. If it fits in no open bin, a new bin is opened.	**The heuristic algorithm that applies the list-processing algorithm to the priority list obtained by listing the tasks in decreasing order of their time length.**
An algorithm that is fast to carry out but that doesn't necessarily give an optimal solution to an optimization problem.	**A heuristic algorithm for bin packing where the first-fit algorithm is applied to the list of weights sorted so that they appear in decreasing order.**

Chapter 3 **Planning and Scheduling** **Independent tasks**	**Chapter 3** **Planning and Scheduling** **List-processing algorithm**
Chapter 3 **Planning and Scheduling** **Machine scheduling**	**Chapter 3** **Planning and Scheduling** **Next fit (NF)**
Chapter 3 **Planning and Scheduling** **Next-fit decreasing (NFD)**	**Chapter 3** **Planning and Scheduling** **Priority list**
Chapter 3 **Planning and Scheduling** **Processor**	**Chapter 3** **Planning and Scheduling** **Ready task**

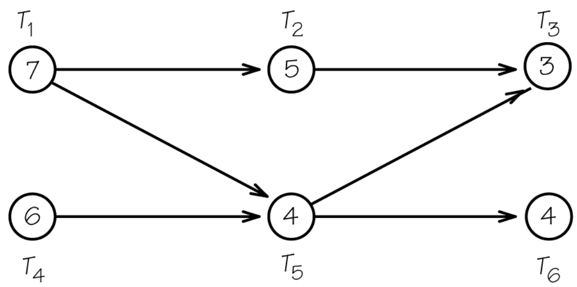
A heuristic algorithm for assigning tasks to processors: Assign the first ready task on the priority list that has not already been assigned to the lowest-numbered processor that is not working on a task.	**Tasks are independent when there are no edges in the order-requirement digraph.**
A heuristic algorithm for bin packing in which a new bin is opened if the weight to be packed next will not fit in the bin that is currently being filled; the current bin is then closed.	**The problem of assigning tasks to processors so as to complete the tasks by the earliest time possible.**
An ordering of the collection of tasks to be scheduled for the purpose of attaining a particular scheduling goal. One such goal is minimizing completion time when the list algorithm is applied.	**A heuristic algorithm for bin packing where the next-fit algorithm is applied to the list of weights sorted so that they appear in decreasing order.**
A task is called ready at a particular time if its predecessors, as given by the order-requirement digraph, have been completed by that time.	**A person, machine, robot, operating room, or runway with time that must be scheduled.**

Chapter 3 **Planning and Scheduling** **Vertex coloring**	**Chapter 3** **Planning and Scheduling** **Worst-case analysis**
Chapter 3 **Planning and Scheduling** **Worst fit (WF)**	**Chapter 3** **Planning and Scheduling** **Worst-fit decreasing (WFD)**

The study of the list-processing algorithm (more generally, any algorithm) from the point of view of how well it performs on the hardest problems it may be used on.	**A vertex coloring of a graph *G* is an assignment of labels, which can be thought of as “colors,” to the vertices of *G* so that vertices joined by an edge get different labels (colors).**
A heuristic algorithm for bin packing where the worst-fit algorithm is applied to the list of weights sorted so that they appear in decreasing order.	**A heuristic algorithm for bin packing in which the next weight to be packed is placed into the open bin with the largest amount of room remaining. If the weight fits in no open bin, a new bin is opened.**

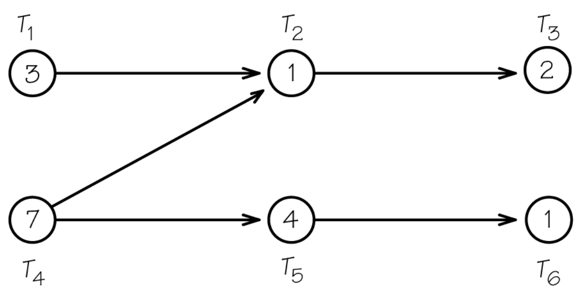
Practice Quiz

1. Given the order-requirement digraph below (with time given in minutes) and the priority list T_1, T_2, T_3, T_4, T_5, T_6, apply the list-processing algorithm to construct a schedule using two processors. How much time is required?



a. 13 minutes
b. 15 minutes
c. 16 minutes

2. Given the order-requirement digraph below (with time given in minutes) and the priority list T_1, T_2, T_3, T_4, T_5, T_6, apply the critical-path scheduling algorithm to construct a schedule using two processors. Which task is scheduled first?



a. T_1
b. T_2
c. T_4

3. The director of a skating show has 25 skaters with varying length numbers to split into three segments, separated by intermissions. This job can be solved using:

a. the list-processing algorithm for independent tasks
b. the critical-path scheduling algorithm
c. the first-fit algorithm for bin packing

4. Use the decreasing-time-list algorithm to schedule the tasks below (times given in minutes) on 2 machines. How much time does the resulting schedule require?

Tasks: 5, 4, 7, 3, 8, 6, 2, 5, 8

a. 24 minutes
b. 25 minutes
c. more than 25 minutes

5. Use the first-fit algorithm to pack the following weights into bins that can hold no more than 10 pounds. How many bins are required?

Weights: 5, 4, 7, 3, 8, 6, 2, 5, 8

a. 5

b. 6

c. more than 6

6. Compare the results of the first-fit and the first-fit-decreasing algorithm to pack the following weights into bins that can hold no more than 10 pounds. Which statement is true?

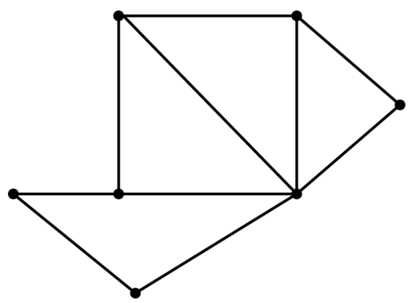
Weights: 5, 4, 7, 3, 8, 6, 2, 5, 8

a. The two algorithms pack the items together in the same way.

b. The two algorithms use the same number of bins, but group the items together in different ways.

c. One algorithm uses fewer bins than the other.

7. Find the chromatic number of the graph below:



a. 5

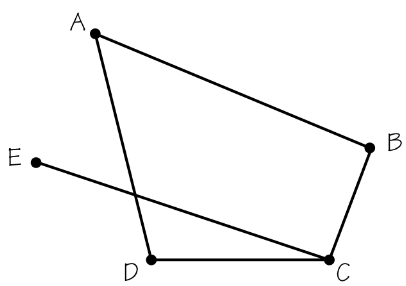
b. 4

c. 3

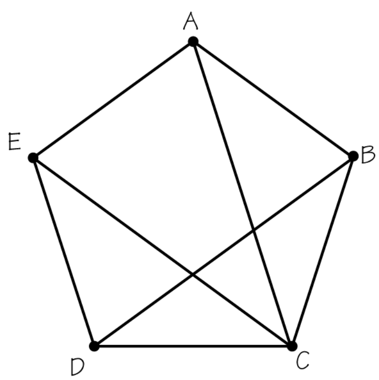
8. An architecture firm must schedule meeting times for its working groups. The following chart indicates which projects have overlapping members for their working groups. Which graph would be used to decide how many different meeting times would be required?

	A	*B*	*C*	*D*	*E*
A		X		X	
B	X		X		
C		X		X	X
D	X		X		
E			X		

a.



b.



c. neither

9. Which of the following statements is true?

I: If the time of a task on the critical path of a digraph is shortened by 3 minutes, overall completion time for the job is shortened by 3 minutes.

II: If additional processors are devoted to a job, the overall time required would be less.

a. I only

b. II only

c. neither I nor II

10. Which of the following statements is true?

I: The worst-fit algorithm for bin packing always produces optimal results.

II: The chromatic number of a graph is always less than the number of vertices of the graph.

a. only I

b. only II

c. neither I nor II

Word Search

Refer to page 107 of your text to obtain the Review Vocabulary. There are 20 hidden vocabulary words/expressions in the word search below. It should be noted that spaces, hyphens, are removed as well as abbreviations. *First fit* and *First-fit decreasing* appear separately. *Next fit* and *Next-fit decreasing* appear separately. Also, *Worse fit* and *Worse-fit decreasing* appear separately.

W	M	Y	P	S	U	V	N	N	K	N	T	S	L	H	H	F	H	Y	E	Q	L	X	Z	R	C	W	R
C	K	R	T	H	I	Q	P	J	E	D	Q	F	N	J	R	K	K	T	I	A	O	T	G	O	I	O	U
R	O	N	Z	Q	C	X	T	E	N	M	A	B	V	E	Y	S	D	D	G	E	A	R	A	A	G	R	K
D	T	M	A	G	W	S	M	X	Z	A	E	I	S	A	S	H	S	Y	E	G	Q	L	L	M	W	S	R
R	T	E	O	J	D	I	E	L	E	W	T	J	S	S	G	S	P	X	K	N	H	N	I	A	D	T	T
K	A	M	S	T	I	F	T	S	R	I	F	V	F	V	S	B	F	D	P	C	A	D	S	R	N	F	K
D	F	V	G	Y	S	M	S	C	T	G	N	I	S	A	E	R	C	E	D	T	I	F	T	S	R	I	F
I	M	A	E	X	W	S	H	H	L	Z	N	P	T	B	F	B	F	C	N	R	T	D	P	I	H	T	S
M	H	T	I	R	O	G	L	A	T	S	I	L	E	M	I	T	G	N	I	S	A	E	R	C	E	D	E
L	T	S	S	S	A	C	T	D	A	W	C	H	E	O	O	T	F	R	O	D	G	C	O	Q	K	E	R
O	I	L	E	F	V	G	B	I	G	N	I	L	U	D	E	H	C	S	E	N	I	H	C	A	M	C	C
O	F	I	N	D	E	P	E	N	D	E	N	T	T	A	S	K	S	B	S	L	O	K	E	S	G	R	W
Q	T	E	W	O	R	S	T	C	A	S	E	A	N	A	L	Y	S	I	S	T	Y	E	S	M	D	E	D
G	X	W	C	S	G	N	I	S	A	E	R	C	E	D	T	I	F	T	X	E	N	L	S	F	L	A	R
O	E	C	D	E	H	E	U	R	I	S	T	I	C	A	L	G	O	R	I	T	H	M	I	O	Z	S	B
T	N	G	N	I	R	O	L	O	C	X	E	T	R	E	V	L	R	I	S	L	S	F	N	S	V	I	E
R	I	I	E	P	N	G	E	C	E	E	I	A	A	E	E	E	Y	H	E	F	C	X	G	R	A	N	Q
D	L	T	D	T	B	I	N	P	A	C	K	I	N	G	P	R	O	B	L	E	M	Y	A	R	L	G	I
N	Q	L	R	U	C	R	I	T	I	C	A	L	P	A	T	H	S	C	H	E	D	U	L	I	N	G	A
T	R	M	Q	S	H	T	J	R	S	F	S	S	R	A	L	S	R	R	D	G	A	I	G	F	Q	C	O
G	O	R	O	N	M	F	N	A	N	V	G	K	S	A	T	Y	D	A	E	R	D	Q	O	S	T	O	N
Y	H	T	M	T	D	M	A	E	N	S	S	E	L	F	A	T	S	I	L	Y	T	I	R	O	I	R	P
G	H	S	L	O	N	P	L	B	Q	J	P	R	M	R	E	O	O	I	A	P	B	E	I	O	F	B	E
T	F	A	I	L	S	A	D	A	K	C	Q	U	F	P	R	O	C	E	S	S	O	R	T	E	T	O	E
L	W	F	E	L	E	F	E	N	B	R	E	B	M	U	N	C	I	T	A	M	O	R	H	C	S	E	S
N	N	O	I	A	P	B	S	L	S	E	K	R	D	E	E	U	E	G	L	V	E	S	M	T	R	A	S
M	S	F	C	N	S	M	A	S	G	S	H	I	T	J	Q	I	E	M	E	D	P	N	A	H	O	R	A
H	Y	R	I	R	R	T	I	E	C	T	E	N	E	W	R	O	E	O	E	Q	T	M	E	E	W	D	D

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

6. ______________________________

7. ______________________________

8. ______________________________

9. ______________________________

10. ______________________________

11. ______________________________

12. ______________________________

13. ______________________________

14. ______________________________

15. ______________________________

16. ______________________________

17. ______________________________

18. ______________________________

19. ______________________________

20. ______________________________