Chapter 3
Planning and Scheduling

Solutions

Exercises:

1. (a) Scheduling final examinations, the times for course being offered, cleaning of classrooms, etc.

(b) Scheduling the buses, the selling of tickets, cleaning, etc.

(c) Scheduling regular staff and supervisory staff, deliveries of items for sale, etc.

(d) Scheduling meal times, shopping, etc.

(e) Scheduling staff, maintenance of backup equipment, etc.

(f) Scheduling nurses, doctors, operating rooms, medical imaging, etc.

(g) Scheduling maintenance and repair of hoses and equipment, personnel to respond to alarms.

(h) Scheduling store personnel, inventory work, shelving new books, and restoring books to proper order on the shelves.

(i) Scheduling ground and squad car patrols around the clock, officers on surveillance assignments, general work involving nonemergency interaction with the public; arrangements must be made regarding who is on call for emergency duty.

3. Jocelyn must perhaps launder her clothes, arrange care for her cat, pack, arrange for a taxi to the airport, and get to the airport. Unless she can get a friend to help her with some of these tasks, she must do all the tasks herself. She can launder her clothes during the time she arranges for a taxi, but most of the tasks cannot be done simultaneously.

5. (a) i. Processor 1: T_1 from 0 to 13, T_3 from 13 to 25, T_6 from 25 to 45.
Processor 2: T_2 from 0 to 18, T_4 from 18 to 27, T_5 from 27 to 35, idle from 35 to 45.

ii. Processor 1: T_1 from 0 to 13, T_3 from 13 to 25, T_4 from 25 to 34, T_5 from 34 to 42.
Processor 2: T_2 from 0 to 18, T_6 from 18 to 38, idle from 38 to 42.

(b) The schedule produced in (ii) is optimal, because the sum of the task times is 80 and no set of tasks can be arranged that will feasibly sum to 40 on each processor.

(c) The critical path is T_2, T_6, and it has length 38. No schedule can be completed by time 38 on two processors because the sum of the task times divided by 2 is 40.

7. (a) Processor 1: T_1, T_2, T_3, T_5, T_7.
Processor 2: Idle 0 to 2, T_4, T_6, idle 4 to 5.

(b) Processor 1: T_1, T_2, T_3, T_6, T_7.
Processor 2: Idle 0 to 2, T_4, T_5, idle 4 to 5.

(c) Yes.

(d) No.

(e) T_3 and T_5.

9. (a) Processor 1: T_1, T_6, idle 15 to 21, T_7, idle 27 to 31.
Processor 2: T_2, T_5, T_8.
Processor 3: T_3, T_4, idle from 13 to 31.

(b) Processor 1: T_1, T_6, idle 15 to 21, T_7, idle 27 to 31.
Processor 2: T_3, T_4, idle from 13 to 21, T_8.
Processor 3: T_2, T_5, idle from 21 to 31.

(c) Processor 1: T_4, idle 10 to 11, T_6, idle 18 to 21, T_8.
Processor 2: T_2, T_5, T_7, idle 27 to 31.
Processor 3: T_1, T_3, idle 11 to 31.

11. Examples include inserting identical mirror systems on different models of cars and vaccinating different children against polio.

13. T_1, T_2, T_3, T_4, T_8, T_9, T_{10}, T_{11}, T_5, T_6, T_7, T_{12}.

15. (a) No. Consider the tasks that begin after the stretch where all machines are idle. Pick one of these tasks *T* and say machine 1 was the machine that it was given to. This task was ready for machine 1 just prior to when it began *T* because no task was just being completed on any other machine at this time because they were all idle. Thus, *T* should have begun earlier on machine 1.

(b) This schedule cannot arise using the list-processing algorithm, because T_2 should have been scheduled at time 0.

(c) Use the digraph with no edges and the list: T_2, T_1, T_3, T_4, T_5.

17. (a) T_1, T_2, T_3, and T_6 are ready at time 0.

(b) No tasks require that T_1 and T_6 be done before these other tasks can begin.

(c) The critical path consists only of T_6 and has length 20.

(d) Processor 1: T_1, T_6: Processor 2: T_2, T_4, idle from 18 to 30: Processor 3: T_3, T_5, idle from 12 to 30.

(e) No.

(f) Processor 1: T_6, idle from 20 to 22: Processor 2: T_3, T_5, T_1: Processor 3: T_2, T_4, idle from 18 to 22.

(g) Yes.

(h) Another list leading to the same optimal schedule is T_6, T_3, T_2, T_4, T_5, T_1.

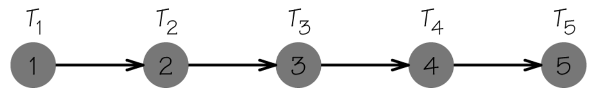
19. (a) 5! = 120

(b) No. Whatever list is used, T_1 must be assigned to the first machine at time 0 because it is the only task ready at time 0.

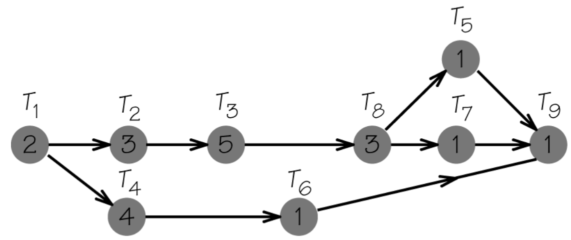
(c) No. First, while Processor 1 works on T_1. Processor 2 must be idle. Second, the task times are integers with sum 31. If there are two processors, one of the processors must have idle time since when 2 divides 31, there is a remainder of 1.

(d) No.

21. Using the order-requirement digraph shown and any list with one or more processors yields the same schedule:



23. (a) One reasonable possibility is (time in min):



The earliest completion time is 15.

(b) The decreasing-time list is T_3, T_4, T_2, T_8, T_1, T_5, T_6, T_7, T_9. The schedule is Processor 1: T_1, T_4, T_6, idle 7 to 10, T_8, T_5, T_9; Processor 2: idle 0 to 2, T_2, T_3, idle 10 to 13, T_7, idle 14 to 15.

25. No. At time 11, T_4 should been assigned to Machine 1 because Machine 1 was free at this time and T_4 was ready.

27. (a) Task times: $T_1 = 3$, $T_2 = 3$, $T_3 = 2$, $T_4 = 3$, $T_5 = 3$, $T_6 = 4$, $T_7 = 5$, $T_8 = 3$, $T_9 = 2$, $T_{10} = 1$, $T_{11} = 1$, and $T_{12} = 3$. This schedule would be produced from the list: T_1, T_3, T_2, T_5, T_4, T_6, T_7, T_8, T_{11}, T_{12}, T_9, T_{10}.

(b) Task times: $T_1 = 3$, $T_2 = 3$, $T_3 = 3$, $T_4 = 2$, $T_5 = 2$, $T_6 = 4$, $T_7 = 3$, $T_8 = 5$, $T_9 = 8$, $T_{10} = 4$, $T_{11} = 7$, $T_{12} = 9$, and $T_{13} = 3$. This schedule would be produced from the list: T_1, T_5, T_7, T_4, T_3, T_6, T_{11}, T_8, T_{12}, T_9, T_2, T_{10}, T_{13}.

29. (a) (i) Processor 1: T_1, T_3, T_5, T_7, idle from 16 to 20; Processor 2: T_2, T_4, T_6, T_8.

(ii) Processor 1: T_8, T_5, T_4, T_1; Processor 2: T_7, T_6, T_3, T_2.

(b) The schedule in (ii) is optimal.

31. Such criteria include decreasing length of the times of the tasks, order of size of financial gains when each task is finished, and increasing length of the times of the tasks.

33. (a) Machine 1: T_1, T_6, T_{10}, idle from 8 to 9; Machine 2: T_3, T_4, T_{11}, T_{12}; Machine 3: T_2, T_7, idle from 8 to 9; Machine 4: T_5, T_8, T_9, idle from 8 to 9.

(b) Machine 1: T_1, T_8, T_{10}; Machine 2: T_5, T_6, T_2, T_{13}; Machine 3: T_7, T_{12}; Machine 4: T_4, T_{11}, idle from 9 to 12; Machine 5: T_3, T_9, idle from 11 to 12.

The schedule in part (a) had to have idle time because the total task time was 33, and 33 is not exactly divisible by 4. The schedule in part (b) had to have idle time because the total task time was 56, and 56 is not exactly divisible by 5.

35. (a) List for (i) yields (with items coded by task time): Machine 1: 12, 9, 15, idle from 36 to 50; Machine 2: 7, 10, 13, 20. List for (ii) yields: Machine 1: 12, 13, 20; Machine 2: 7, 9, 15, 10, idle from 41 to 45. List for (iii) yields: Machine 1: 20, 12, 9, idle from 41 to 45; Machine 2: 15, 13, 10, 7.

(b) These schedules complete earlier than those where precedence constraints hold. An optimal schedule is possible, however: Machine 1: 20, 10, 13; Machine 2: 12, 15, 9, 7. The associated list is: T_6, T_1, T_5, T_7, T_2, T_4, T_3.

(c) The critical path list is T_6, T_5, T_4, T_1, T_7, T_2, T_3 using the first processor. It finishes at time 41 and is idle until 45, when processor 2 finishes.

37. (a) The tasks are scheduled on the machines as follows: Processor 1: 12, 13, 45, 34, 63, 43, 16, idle 226 to 298; Processor 2: 23, 24, 23, 53, 25, 74, 76; Processor 3: 32, 23, 14, 21, 18, 47, 23, 43, 16, idle 237 to 298.

(b) The tasks are scheduled on the machines as follows: Processor 1: 12, 24, 14, 34, 25, 23, 16, 16, 76; Processor 2: 23, 23, 21, 63, 43, idle 173 to 240; Processor 3: 32, 23, 53, 74, idle 182 to 240; Processor 4: 13, 45, 18, 47, 43, idle 166 to 240.

(c) The decreasing-time list is 76, 74, 63, 53, 47, 45, 43, 43, 34, 32, 25, 24, 23, 23, 23, 23, 21, 18, 16, 16, 14, 13, 12. The tasks are scheduled on three machines as follows: Processor 1: 76, 45, 43, 24, 23, 18, 16, 13; Processor 2: 74, 47, 34, 32, 23, 21, 14, 12, idle 257 to 258; Processor 3: 63, 53, 43, 25, 23, 23, 16, idle 246 to 248. The tasks are scheduled on four machines as follows: Processor 1: 76, 43, 24, 23, 16, idle 182 to 194; Processor 2: 74, 43, 25, 23, 16, 13; Processor 3: 63, 45, 32, 23, 18, 12, idle 193 to 194; Processor 4: 53, 47, 34, 23, 21, 14, idle 192 to 194.

(d) The new decreasing time list is 84, 82, 71, 61, 55, 45, 43, 43, 34, 32, 25, 24, 23, 23, 23, 23, 21, 18, 16, 16, 14, 13, 12. The tasks are scheduled as follows: Processor 1: 84, 45, 43, 25, 23, 23, 16, 12; Processor 2: 82, 55, 34, 32, 23, 18, 14, 13; Processor 3: 71, 61, 43, 24, 23, 21, 16, idle 259 to 271.

39. Examples include jobs in a videotape copying shop, data entry tasks in a computer system, and scheduling nonemergency operations in an operating room. These situations may have tasks with different priorities, but there is no physical reason for the tasks not to be independent, as would be the case with putting on a roof before a house had walls erected.

41. Each task heads a path of length equal to the time to do that task.

43. The times to photocopy the manuscripts, in decreasing order, are 120, 96, 96, 88, 80, 76, 64, 64, 60, 60, 56, 48, 40, 32. Packing these in bins of size 120 yields Bin 1: 120; Bin 2: 96; Bin 3: 96; Bin 4: 88, 32; Bin 5: 80, 40; Bin 6: 76; Bin 7: 64, 56; Bin 8: 64, 48; Bin 9: 60, 60. Nine photocopy machines are needed to finish within 2 minutes using FFD. The number of bins would not change, but the placement of the items in the bins would differ for worst-fit decreasing.

45. (a) Using the next-fit algorithm, the bins are filled as follows: Bin 1: 12, 15; Bin 2: 16, 12; Bin 3: 9, 11, 15; Bin 4: 17, 12; Bin 5: 14, 17; Bin 6: 18; Bin 7: 19; Bin 8: 21; Bin 9: 31; Bin 10: 7, 21; Bin 11: 9, 23; Bin 12: 24; Bin 13: 15, 16; Bin 14: 12, 9, 8; Bin 15: 27; Bin 16: 22; Bin 17: 18.

(b) The decreasing list is 31, 27, 24, 23, 22, 21, 21, 19, 18, 18, 17, 17, 16, 16, 15, 15, 15, 14, 12, 12, 12, 12, 11, 9, 9, 9, 8, 7. The next-fit decreasing schedule is Bin 1: 31; Bin 2: 27; Bin 3: 24; Bin 4: 23; Bin 5: 22; Bin 6: 21; Bin 7: 21; Bin 8: 19; Bin 9: 18, 18; Bin 10: 17, 17; Bin 11: 16, 16; Bin 12: 15, 15; Bin 13: 15, 14; Bin 14: 12, 12, 12; Bin 15: 12, 11, 9; Bin 16: 9, 9, 8, 7.

Continued on next page

45. continued

(c) The worst-fit schedule using the original list is Bin 1: 12, 15, 9; Bin 2: 16, 12; Bin 3: 11, 15; Bin 4: 17, 12; Bin 5: 14, 17; Bin 6: 18, 7; Bin 7: 19, 9; Bin 8: 21, 15; Bin 9: 31; Bin 10: 21, 9; Bin 11: 23, 8; Bin 12: 24; Bin 13: 16, 12; Bin 14: 27; Bin 15: 22; Bin 16: 18.

(d) The worst-fit decreasing schedule would be Bin 1: 31; Bin 2: 27, 9; Bin 3: 24, 12; Bin 4: 23, 12; Bin 5: 22, 14; Bin 6: 21, 15; Bin 7: 21, 15; Bin 8: 19, 17; Bin 9: 18, 18; Bin 10: 17, 16; Bin 11: 16, 15; Bin 12: 12, 12, 11; Bin 13: 9, 9, 8, 7.

47. The bins have a capacity of 120. (First fit): Bin 1: 63, 32, 11; Bin 2: 19, 24, 64; Bin 3: 87, 27; Bin 4: 36, 42; Bin 5: 63. This schedule would take five station breaks; however, the total time for the breaks is under 8 minutes. The decreasing list is 87, 64, 63, 63, 42, 36, 32, 27, 24, 19, 11. (First-fit decreasing): Bin 1: 87, 32; Bin 2: 64, 42, 11; Bin 3: 63, 36, 19; Bin 4: 63, 27, 24. This solution uses only four station breaks.

49. (a) There are theoretical results that show that best fit "usually" performs better than worst fit.

(b) Try 8, 7, 5, 3, 3, 2 in bins of capacity 14.

51. The total performance time exceeds what will fit on four disks. Using FFD, one can fit the music on five disks.

53. For problems with few weights to be packed, small integer weights, and a small integer as bin capacity, this method can work well. However, when these special conditions are not met, it is very time-consuming to carry out this method. For example, imagine trying to use this method for 2000 random real numbers of the form *.xyz*, where *x*, *y*, and *z* are decimal digits and the bin capacity is 1.

55. It makes sense to leave bins open as more items arrive to be packed if the cost of having many bins open at once is reasonable and there is room to have many partially-filled bins open without incurring great inconvenience or cost. One such example might be a company that has room for many identical trucks to park as they are loaded with goods to be delivered. There may be complex cost trade-offs between sending off fewer trucks because we wait to pack as much into each truck as possible and sending out more partially-filled trucks.

57. (a) The schedule with four secretaries is as follows: Processor 1: 25, 36, 15, 15, 19, 15, 27; Processor 2: 18, 32, 18, 31, 30, 18; Processor 3: 13, 30, 17, 12, 18, 16, 16, 16, 14; Processor 4: 19, 12, 25, 26, 18, 12, 24, 9.

The schedule with five secretaries is as follows: Processor 1: 25, 25, 31, 12, 16, 14; Processor 2: 18, 12, 17, 12, 15, 30, 9; Processor 3: 13, 32, 26, 16, 15, 18; Processor 4: 19, 36, 18, 19, 24; Processor 5: 30, 18, 15, 18, 16, 27.

(b) The decreasing time list is 36, 32, 31, 30, 30, 27, 26, 25, 25, 24, 19, 19, 18, 18, 18, 18, 18, 17, 16, 16, 16, 15, 15, 15, 14, 13, 12, 12, 12, 9.

The schedule using this list on four processors would be Processor 1: 36, 25, 19, 18, 17, 16, 13, 9; Processor 2: 32, 26, 25, 18, 16, 15, 12; Processor 3: 31, 27, 24, 18, 16, 15, 12, 12; Processor 4: 30, 30, 19, 18, 18, 15, 14.

The schedule using this list on five processors would be Processor 1: 36, 24, 18, 16, 14, 12; Processor 2: 32, 25, 18, 18, 15, 12; Processor 3: 31, 25, 19, 18, 15, 9; Processor 4: 30, 27, 18, 17, 15, 13; Processor 5: 30, 26, 19, 16, 16, 12.

(c) The five-processor decreasing-time schedule is optimal (time 120), but the four-processor decreasing-time schedule is not. One can see this, since when the task of length 17 scheduled on processor 1 and the task of length 18 on processor 3 are interchanged, the completion time is reduced to 154 from 155 for the four-processor, decreasing-time schedule.

Continued on next page

57. continued

(d) As a bin-packing problem, each bin will have a capacity of 60. Using the decreasing list, we obtain the following packings: (First-fit decreasing): Bin 1: 36, 24; Bin 2: 32, 27; Bin 3: 31, 26; Bin 4: 30, 30; Bin 5: 25, 25, 9; Bin 6: 19, 19, 18; Bin 7: 18, 18, 18; Bin 8: 18, 17, 16; Bin 9: 16, 16, 15, 13; Bin 10: 15, 15, 14, 12; Bin 11: 12, 12.

(e) NFD uses 13 bins. Bin 1: 36; Bin 2: 32; Bin 3: 31; Bin 4: 30, 30; Bin 5: 27, 26; Bin 6: 25, 25; Bin 7: 24, 19; Bin 8: 19, 18, 18; Bin 9: 18, 18, 18; Bin 10: 17, 16, 16; Bin 11: 16, 15, 15; Bin 12: 15, 14, 13, 12; Bin 13: 12, 12, 9. WFD uses 11 bins. Bin 1: 36, 24; Bin 2: 32, 26; Bin 3: 31, 27; Bin 4: 30, 30; Bin 5: 25, 25; Bin 6: 19, 19, 18; Bin 7: 18, 18, 18; Bin 8: 18, 17, 16; Bin 9: 16, 16, 15, 13; Bin 10: 15, 15, 14, 12; Bin 11: 12, 12, 9.

(f) An optimal packing with 10 bins exists. Bin 1: 36, 24; Bin 2: 32, 16, 12; Bin 3: 31, 17, 12; Bin 4: 30, 30; Bin 5: 27, 18, 15; Bin 6: 26, 18, 16; Bin 7: 25, 19, 16; Bin 8: 25, 19, 15; Bin 9: 18, 18, 12, 9; Bin 10: 18, 15, 14, 13.

59. (a) Packing boxes of the same height into crates: packing want ads into a newspaper page.

(b) We assume, without loss of generality, $p \geq q$. One heuristic, similar to first-fit, orders the rectangles $p \times q$ as in a dictionary (i.e., $p \times q$ listed prior to $r \times s$ if $p > r$ or $p = r$ and $q \geq s$). It then puts the rectangles in place in layers in a first-fit manner; that is, do not put a rectangle into a second layer until all positions on the first layer are filled. However, extra room in the first layer is “wasted.”

(c) The problem of packing rectangles of width 1 in an $m \times 1$ rectangle is a special case of the two-dimensional problem, equivalent to the bin-packing problem we have discussed.

(d) Two 1×10 rectangles cannot be packed into a 5×4 rectangle, even though there would be an area of 20 in this rectangle.

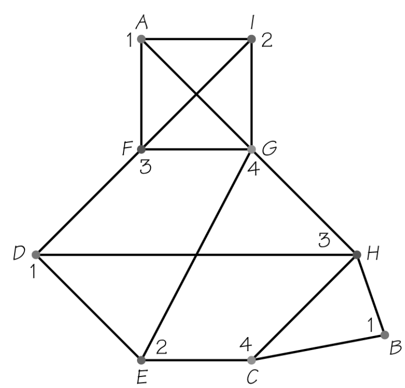
61. There is an example of a bin-packing problem for which a given list takes a certain number of bins, and when an item is deleted from the list, more bins are required. In this example, the deleted item is not first in the list.

63. (a) Graphs (a), (d), (e), and (f) can be colored with three colors, but graphs (b) and (c) cannot.

(b) Graphs (a), (b), (d), (e), and (f) can be colored with four colors, but graph (c) cannot.

(c) The chromatic number for graphs (a) through (f) are, respectively, 3, 4, 5, 2, 3, and 2.

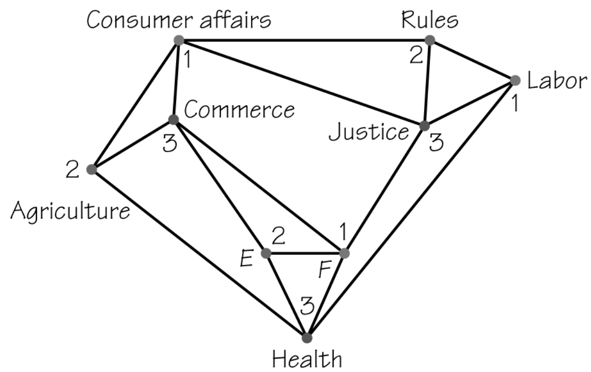
65. (a) Construct the graph shown below:



The vertices of this graph can be colored with no fewer than four colors (1, 2, 3, 4 are used to denote the colors in the figure). Hence, four tanks can be used to display the fish.

(b) The coloring in (a) shows that one can display two types of fish in three of the tanks, and three types of fish in one tank. Since 4 does not divide 9, one cannot do better.

67. (a) The graph for this situation is shown below. The vertices can be labeled with the colors 1, 2, 3 as shown.



(b) Since the vertices can be colored with three colors (and no fewer), the minimum number of time slots for scheduling the committees is three.

(c) The committees can be scheduled in three rooms during each time slot. This might be significant if there were only three rooms that had microphone systems.

69. (a) Three time slots. To solve this problem, draw a graph by joining the vertices representing two committees if there is no x in the row and column of the table for these two committees.

(b) It is possible to three-color this graph so that each of the three colors is used three times. This means that one needs three rooms to arrange the scheduling of the nine committees.

71. Start at any vertex of the tree and label this vertex with color 1; color any vertex attached by an edge to this vertex with color 2. Continue to color the vertices in the tree in this manner, alternating the use of colors. If some vertex were attached to both a vertex colored 1 and another vertex colored 2, at some stage this would imply the graph had a circuit (of odd length), which is not possible, since trees have no circuits of any length.

73. The edge-coloring numbers for graphs (a) through (f) of Exercise 63 are, respectively, 6, 8, 6, 3, 3, and 4. The minimum edge-coloring number of any graph is either the maximal valence of any vertex in the graph or one more than the maximal valence. (This fact was discovered by the Russian mathematician Vizing.)

75. (a) Graph (a) four colors; graph (b) two colors; graph (c) four colors; graph (d) four colors; graph (e) two colors; graph (f) three colors.

(b) Coloring the maps of countries in an atlas would be one application of face colorings of graphs.

77. 3