



FIGURE 2 Showing that the Halting Problem is Unsolvable.

is “halt,” then by the definition of K we see that $K(K)$ loops forever, in violation of what H tells us. In both cases, we have a contradiction.

Thus, H cannot always give the correct answers. Consequently, there is no procedure that solves the halting problem. ◀

Exercises

- List all the steps used by Algorithm 1 to find the maximum of the list 1, 8, 12, 9, 11, 2, 14, 5, 10, 4.
- Determine which characteristics of an algorithm described in the text (after Algorithm 1) the following procedures have and which they lack.
 - procedure** *double*(n : positive integer)
`while` $n > 0$
 $n := 2n$
 - procedure** *divide*(n : positive integer)
`while` $n \geq 0$
 $m := 1/n$
 $n := n - 1$
 - procedure** *sum*(n : positive integer)
`sum` := 0
`while` $i < 10$
 $sum := sum + i$
 - procedure** *choose*(a, b : integers)
 $x :=$ either a or b
- Devise an algorithm that finds the sum of all the integers in a list.
- Describe an algorithm that takes as input a list of n integers and produces as output the largest difference obtained by subtracting an integer in the list from the one following it.
- Describe an algorithm that takes as input a list of n integers in nondecreasing order and produces the list of all values that occur more than once. (Recall that a list of integers is **nondecreasing** if each integer in the list is at least as large as the previous integer in the list.)
- Describe an algorithm that takes as input a list of n integers and finds the number of negative integers in the list.
- Describe an algorithm that takes as input a list of n integers and finds the location of the last even integer in the list or returns 0 if there are no even integers in the list.
- Describe an algorithm that takes as input a list of n distinct integers and finds the location of the largest even integer in the list or returns 0 if there are no even integers in the list.
- A **palindrome** is a string that reads the same forward and backward. Describe an algorithm for determining whether a string of n characters is a palindrome.
- Devise an algorithm to compute x^n , where x is a real number and n is an integer. [Hint: First give a procedure for computing x^n when n is nonnegative by successive multiplication by x , starting with 1. Then extend this procedure, and use the fact that $x^{-n} = 1/x^n$ to compute x^n when n is negative.]
- Describe an algorithm that interchanges the values of the variables x and y , using only assignments. What is the minimum number of assignment statements needed to do this?
- Describe an algorithm that uses only assignment statements that replaces the triple (x, y, z) with (y, z, x) . What is the minimum number of assignment statements needed?
- List all the steps used to search for 9 in the sequence 1, 3, 4, 5, 6, 8, 9, 11 using
 - a linear search.
 - a binary search.
- List all the steps used to search for 7 in the sequence given in Exercise 13 for both a linear search and a binary search.
- Describe an algorithm that inserts an integer x in the appropriate position into the list a_1, a_2, \dots, a_n of integers that are in increasing order.
- Describe an algorithm for finding the smallest integer in a finite sequence of natural numbers.
- Describe an algorithm that locates the first occurrence of the largest element in a finite list of integers, where the integers in the list are not necessarily distinct.
- Describe an algorithm that locates the last occurrence of the smallest element in a finite list of integers, where the integers in the list are not necessarily distinct.

19. Describe an algorithm that produces the maximum, median, mean, and minimum of a set of three integers. (The **median** of a set of integers is the middle element in the list when these integers are listed in order of increasing size. The **mean** of a set of integers is the sum of the integers divided by the number of integers in the set.)
 20. Describe an algorithm for finding both the largest and the smallest integers in a finite sequence of integers.
 21. Describe an algorithm that puts the first three terms of a sequence of integers of arbitrary length in increasing order.
 22. Describe an algorithm to find the longest word in an English sentence (where a sentence is a sequence of symbols, either a letter or a blank, which can then be broken into alternating words and blanks).
 23. Describe an algorithm that determines whether a function from a finite set of integers to another finite set of integers is onto.
 24. Describe an algorithm that determines whether a function from a finite set to another finite set is one-to-one.
 25. Describe an algorithm that will count the number of 1s in a bit string by examining each bit of the string to determine whether it is a 1 bit.
 26. Change Algorithm 3 so that the binary search procedure compares x to a_m at each stage of the algorithm, with the algorithm terminating if $x = a_m$. What advantage does this version of the algorithm have?
 27. The **ternary search algorithm** locates an element in a list of increasing integers by successively splitting the list into three sublists of equal (or as close to equal as possible) size, and restricting the search to the appropriate piece. Specify the steps of this algorithm.
 28. Specify the steps of an algorithm that locates an element in a list of increasing integers by successively splitting the list into four sublists of equal (or as close to equal as possible) size, and restricting the search to the appropriate piece.
- In a list of elements the same element may appear several times. A **mode** of such a list is an element that occurs at least as often as each of the other elements; a list has more than one mode when more than one element appears the maximum number of times.
29. Devise an algorithm that finds a mode in a list of nondecreasing integers. (Recall that a list of integers is nondecreasing if each term is at least as large as the preceding term.)
 30. Devise an algorithm that finds all modes. (Recall that a list of integers is nondecreasing if each term of the list is at least as large as the preceding term.)
 31. Devise an algorithm that finds the first term of a sequence of integers that equals some previous term in the sequence.
 32. Devise an algorithm that finds all terms of a finite sequence of integers that are greater than the sum of all previous terms of the sequence.
 33. Devise an algorithm that finds the first term of a sequence of positive integers that is less than the immediately preceding term of the sequence.
 34. Use the bubble sort to sort 6, 2, 3, 1, 5, 4, showing the lists obtained at each step.
 35. Use the bubble sort to sort 3, 1, 5, 7, 4, showing the lists obtained at each step.
 36. Use the bubble sort to sort d, f, k, m, a, b , showing the lists obtained at each step.
 - *37. Adapt the bubble sort algorithm so that it stops when no interchanges are required. Express this more efficient version of the algorithm in pseudocode.
 38. Use the insertion sort to sort the list in Exercise 34, showing the lists obtained at each step.
 39. Use the insertion sort to sort the list in Exercise 35, showing the lists obtained at each step.
 40. Use the insertion sort to sort the list in Exercise 36, showing the lists obtained at each step.
- The **selection sort** begins by finding the least element in the list. This element is moved to the front. Then the least element among the remaining elements is found and put into the second position. This procedure is repeated until the entire list has been sorted.
41. Sort these lists using the selection sort.

a) 3, 5, 4, 1, 2	b) 5, 4, 3, 2, 1
c) 1, 2, 3, 4, 5	
 42. Write the selection sort algorithm in pseudocode.
 -  43. Describe an algorithm based on the linear search for determining the correct position in which to insert a new element in an already sorted list.
 44. Describe an algorithm based on the binary search for determining the correct position in which to insert a new element in an already sorted list.
 45. How many comparisons does the insertion sort use to sort the list $1, 2, \dots, n$?
 46. How many comparisons does the insertion sort use to sort the list $n, n - 1, \dots, 2, 1$?
- The **binary insertion sort** is a variation of the insertion sort that uses a binary search technique (see Exercise 44) rather than a linear search technique to insert the i th element in the correct place among the previously sorted elements.
47. Show all the steps used by the binary insertion sort to sort the list 3, 2, 4, 5, 1, 6.
 48. Compare the number of comparisons used by the insertion sort and the binary insertion sort to sort the list 7, 4, 3, 8, 1, 5, 4, 2.
 - *49. Express the binary insertion sort in pseudocode.
 50. a) Devise a variation of the insertion sort that uses a linear search technique that inserts the j th element in the correct place by first comparing it with the $(j - 1)$ st element, then the $(j - 2)$ th element if necessary, and so on.
 - b) Use your algorithm to sort 3, 2, 4, 5, 1, 6.
 - c) Answer Exercise 45 using this algorithm.
 - d) Answer Exercise 46 using this algorithm.

51. When a list of elements is in close to the correct order, would it be better to use an insertion sort or its variation described in Exercise 50?
52. Use the greedy algorithm to make change using quarters, dimes, nickels, and pennies for
- a) 87 cents. b) 49 cents.
c) 99 cents. d) 33 cents.
53. Use the greedy algorithm to make change using quarters, dimes, nickels, and pennies for
- a) 51 cents. b) 69 cents.
c) 76 cents. d) 60 cents.
54. Use the greedy algorithm to make change using quarters, dimes, and pennies (but no nickels) for each of the amounts given in Exercise 52. For which of these amounts does the greedy algorithm use the fewest coins of these denominations possible?
55. Use the greedy algorithm to make change using quarters, dimes, and pennies (but no nickels) for each of the amounts given in Exercise 53. For which of these amounts does the greedy algorithm use the fewest coins of these denominations possible?
56. Show that if there were a coin worth 12 cents, the greedy algorithm using quarters, 12-cent coins, dimes, nickels, and pennies would not always produce change using the fewest coins possible.
57. Use Algorithm 7 to schedule the largest number of talks in a lecture hall from a proposed set of talks, if the starting and ending times of the talks are 9:00 A.M. and 9:45 A.M.; 9:30 A.M. and 10:00 A.M.; 9:50 A.M. and 10:15 A.M.; 10:00 A.M. and 10:30 A.M.; 10:10 A.M. and 10:25 A.M.; 10:30 A.M. and 10:55 A.M.; 10:15 A.M. and 10:45 A.M.; 10:30 A.M. and 11:00 A.M.; 10:45 A.M. and 11:30 A.M.; 10:55 A.M. and 11:25 A.M.; 11:00 A.M. and 11:15 A.M.
58. Show that a greedy algorithm that schedules talks in a lecture hall, as described in Example 7, by selecting at each step the talk that overlaps the fewest other talks, does not always produce an optimal schedule.
- *59. a) Devise a greedy algorithm that determines the fewest lecture halls needed to accommodate n talks given the starting and ending time for each talk.
b) Prove that your algorithm is optimal.

Suppose we have s men m_1, m_2, \dots, m_s and s women w_1, w_2, \dots, w_s . We wish to match each person with a member



of the opposite gender. Furthermore, suppose that each person ranks, in order of preference, with no ties, the people of the opposite gender. We say that a matching of people of opposite genders to form couples is **stable** if we cannot find a man m and a woman w who are not assigned to each other such that m prefers w over his assigned partner and w prefers m to her assigned partner.

60. Suppose we have three men m_1, m_2 , and m_3 and three women w_1, w_2 , and w_3 . Furthermore, suppose that the preference rankings of the men for the three women, from highest to lowest, are $m_1: w_3, w_1, w_2$; $m_2: w_1, w_2, w_3$; $m_3: w_2, w_3, w_1$; and the preference rankings of the women for the three men, from highest to lowest, are $w_1: m_1, m_2, m_3$; $w_2: m_2, m_1, m_3$; $w_3: m_3, m_2, m_1$. For each of the six possible matchings of men and women to form three couples, determine whether this matching is stable.

The **deferred acceptance algorithm**, also known as the **Gale-Shapley algorithm**, can be used to construct a stable matching of men and women. In this algorithm, members of one gender are the **suitors** and members of the other gender the **suites**. The algorithm uses a sequence of rounds; in each round every suitor whose proposal was rejected in the previous round proposes to his or her highest ranking suitee who has not already rejected a proposal from this suitor. A suitee rejects all proposals except that from the suitor that this suitee ranks highest among all the suitors who have proposed to this suitee in this round or previous rounds. The proposal of this highest ranking suitor remains pending and is rejected in a later round if a more appealing suitor proposes in that round. The series of rounds ends when every suitor has exactly one pending proposal. All pending proposals are then accepted.

61. Write the deferred acceptance algorithm in pseudocode.
62. Show that the deferred acceptance algorithm terminates.
- *63. Show that the deferred acceptance always terminates with a stable assignment.
64. Show that the problem of determining whether a program with a given input ever prints the digit 1 is unsolvable.
65. Show that the following problem is solvable. Given two programs with their inputs and the knowledge that exactly one of them halts, determine which halts.
66. Show that the problem of deciding whether a specific program with a specific input halts is solvable.

3.2 The Growth of Functions

Introduction

In Section 3.1 we discussed the concept of an algorithm. We introduced algorithms that solve a variety of problems, including searching for an element in a list and sorting a list. In Section 3.3 we will study the number of operations used by these algorithms. In particular, we will estimate the number of comparisons used by the linear and binary search algorithms to find an element in a sequence of n elements. We will also estimate the number of comparisons used by the