
3. Determine the number of character comparisons made by the brute-force algorithm in searching for the pattern GANDHI in the text THERE_IS_MORE_TO_LIFE_THAN_INCREASING_ITS_SPEED (Assume that the length of the text—it is 47 characters long—is known before the search starts.)

4. How many comparisons (both successful and unsuccessful) are made by the brute-force string-matching algorithm in searching for pattern “00001” in the binary text of 1000 zeros?
5 Let $x_1 < x_2 < x_3 \ldots < x_n$ be real numbers representing coordinates of $n$ villages located along a straight road. A post office needs to be built in one of these villages.

Design a brute-force algorithm to find the post-office location minimizing the average distance between the villages and the post office.

Can you reduce the running time to $\Theta(1)$? Explain your answer.

6 (decrease-by-one) A detachment of $n$ soldiers must cross a wide and deep river with no bridge in sight. They notice two 12-year-old boys playing in a rowboat by the shore. The boat is so tiny, however, that it can only hold two boys or one soldier. How can the soldiers get across the river and leave the boys in joint possession of the boat? How many times need the boat pass from shore to shore?
7 Apply insertion sort to sort the list “E X A M P L E” in alphabetical order.

8 Apply DFS to the following tree. Mark out the order in which the nodes are visited.

9 Design a decrease-by-half algorithm for computing $\left\lfloor \log_2 n \right\rfloor$ and determine its time efficiency.
10
a. Write pseudocode for a divide-and-conquer algorithm for finding the position of the largest element in an array of $n$ numbers.

b. What will be your algorithm’s output for arrays with several elements of the largest value?

c. Set up and solve a recurrence relation for the number of key comparisons made by your algorithm.

d. How does this algorithm compare with the brute-force algorithm for this problem?
11 Apply mergesort to sort the list E, X, A, M, P, L, E in alphabetical order. Is mergesort a stable sorting algorithm?