

The time of computer algorithms depends in part on the nature of the memory used. Often simplified models of the memory are used to get at the essence of the situation.

1. Algorithm 1.4 from the text adds the n -digit number x to the n -digit number y and stores the result in the n digit number z . The two numbers and the answer are stored in an array. Suppose the algorithm is run on a computer where no time is needed for computation or references to simple variables, but one unit of time is needed each time an array element is read or stored. (This model might be used to approximate a computer with a fast cache memory that was too small to hold the array elements and a slower memory that was large enough to hold the array elements.) Under these assumptions, how long does Algorithm 1.4 need to add to n -digit numbers?
2. Suppose you want to run Algorithm 1.4 on a computer where array accesses take time proportional to the distance between the current array element and the previous array element that was most recently referenced. In this case, you might want to take great care as you decide where to put each array element.
 - 2a. For the three numbers, x , y , and z , give a formula describing where you will store the i digit of the number. (A program would need this formula to access the digits.) Each digit needs one computer word. You are not permitted to store the z digits on top of the x or y digits. Choose locations so that the program will run fast. Also, draw a picture to help us understand what you intend your memory layout to be.
 - 2b. Analyze the time needed by Algorithm 1.4 when it uses the memory layout that you give in your answer 2a. Assume that if the program has most recently referenced cell number i and is now referencing cell number j then the time needed to reference cell number j is $|i - j|$. Assume one unit of time is needed for the first reference (since there is no previous cell).
3. Give a formula for $W(n)$, the number of computer words needed to hold the positive integer n on a computer that has 32 bit words and a non-negative integer data type (having a non-negative integer data type means that the computer can represent an integer using all of the bits to represent the magnitude of the integer; it does not need one of the bits to represent a sign for the integer).

4. Consider the sum

$$\sum_{0 \leq i \leq k} \binom{n}{i} p^i (1-p)^{n-i}$$

when p is a probability and n is a large integer.

- 4a. What is the value of the sum when $k = n$?
- 4b. What is the index of the term with the largest value when $k = n$?
- 4c. What is the index of the term with the largest value for general k ?
5. Suppose you want to approximate the sum in problem 4 with the sum

$$a \sum_{0 \leq i \leq k} x^i,$$

when k is fairly small.

- 5a. Suppose you want an upper bound on the sum in problem 4. What are good values for a and x ? Explain the reasoning for your answer.
 - 5b. Suppose you want an lower bound on the sum in problem 4. What are good values for a and x ? Explain the reasoning for your answer.
 - 5c. What is the largest value of k for which your reasoning is correct. (Part c of the question means that you can do parts a and b, without worrying a lot about the fact that your reasoning is likely to only work when k is not too large.)
6. Find an asymptotic solution for x when

$$x = t - a \ln x$$

for the case where a is a fixed positive constant and t is large.