Topics : Algorithm Design - Data Structure Design: Data Modeling, Modularity and Information Hiding, Data Abstraction and Encapsulation, Data Representation, Data Types and Abstract Data Types.

Reading :

Topic	Reference	Remarks
Data Abstraction and Abstract Data Types	T1 Sections 2.1 and 2.2	Required

Review Questions

1. What is a Data Model? Why is Data Modeling useful?
2. What is the need for Modularity in product design?
3. What is the principle of Information Hiding?
4. How is Information Hiding related to Data Abstraction?
5. How is Data Encapsulation related to Data Abstraction?
6. What is the purpose of "types"?
7. What is a "data type"? How is it useful?
8. How do you identify a data type?
9. What is an "abstract data type"?
10. Why is data representation important?

Analytical Exercises (AE)

1. Model the data in each of the following examples:
    a. A document is made of paragraphs, each paragraph is made of sentences, each sentence is made of words.
    b. A DNA sequence is a sequence of nucleotide bases. Each base is one of the Adenine, Cytosine, Thymine and Guanine. The characteristics of a sequence are directly inferred from the bases (i.e. no further information is needed.)
    c. An address in India has the following general shape:
        i. A State or Union Territory is the outermost unit.

ii. If the address is in a city/town a postal code (PIN) is enough.
iii. But if it is located in a village, a Taluk enclosing the village, and a District enclosing the Taluk are necessary apart from the postal code.
iv. If the address is in a city/town then a suburb (termed as Nagar, Puram, Pallya, Vihar, Layout, Pet, etc.) is the next enclosing unit within which you either have a main road and a cross road OR a single road name. Depending on the type of the house, either a building number or a building number and an apartment number within the building are included. A landmark is optional.
v. If the address is in a village, then a landmark, and a street name are essential whereas there may or may not be a building number.

d. The name of an individual in India may depend on various cultural characteristics of the individual. Here are some samples:
   i. A name is written as Last Name (i.e. SurName) followed by First Name. The Last Name is always abbreviated as an initial. [Otherwise a random First Name cannot be distinguished from a SurName, because a SurName is the First Name of the father.]
   ii. A name is written as Last Name (i.e. SurName or Family Name) followed by several Names (First/Middle) followed by an optional Titular Name. The Last Name and the First/Middle names may all be abbreviated as initials. Typically the penultimate name (i.e. the ultimate First/Middle name) may not be abbreviated.
   iii. A name is written as Origin Name (i.e. the location of origin of the maternal family), followed by Last Name (i.e. paternally inherited Surname) followed by First Name. The Origin Name and the Last Name are usually abbreviated as initials.
   iv. A name is written as First Name followed by Last Name. If the First Name is long and/or multi-syllabic then it may be abbreviated.
   v. A name is written as First Name followed by a Generic Name (i.e. a very common name shared by almost all

individuals of a certain community) followed by an Origin Name (i.e. the location of origin of the individual)

e. A call log (e.g. missed calls) in a mobile phone.
f. An address book in an email client (say Microsoft Outlook or Mozilla Thunderbird).

2. Analyze the space time tradeoff for the different options proposed in PE1.
3. Analyse the solutions for DP2-a. and DP2-b. - in particular their complexities. [Hint: A PDA is simpler than doing arithmetic. End of Hint.]
4. If you try withdrawing money from a bank in Pilani, you realize that the withdrawal form passes through several clerks/tellers before you receive any money. Each clerk/teller has a list of forms to process and they find that "stacking" them (i.e. placing them in LIFO order) is easier than "queueing" the forms. But they want to be fair and serve customers in FIFO order. Model the counters (i.e. clerks/tellers) using LIFO and FIFO lists to ensure that customers are served in FIFO order as much as possible but clerks/tellers can use "stacking" instead of "queueing" as much as possible.
5. Remodel the history list such that both chronological and reverse-chronological order traversals of the list are supported i.e. *next* as well as *previous* operations are meaningful. (Of course, this applies only for history. Future cannot be predicted :-)
6. Analyse the conditions in parts (b) and (c) of PE4. Argue whether they are equivalent or not.

Design Problem

1. Design an ADT CurrencyMap for the following problem: A currency exchange broker makes profit by executing a sequence of currency exchanges cyclically to exploit the differences in the exchange rates. For this purpose, the broker uses the CurrencyMap which is used for looking up the exchange rate from one currency to another e.g. Indian Rupees (INR) to Chinese Renmimbi (RMB). Assume an appropriate data model.
2.
    a. Use ADT Stack to design an algorithm to verify whether a given string made of parentheses is a valid string of matching

parentheses. A string is a valid string of matching parentheses if every left parenthesis is closed by a corresponding right parenthesis and no right parenthesis occurs before the corresponding left parenthesis.

b. Redo DP2 without using a stack. [Hint: Just use count matching and prefix counts. End of Hint.]
c. Can the solutions to (a) and (b) - separately - be extended for the case where the string is made of matching parentheses, braces, and brackets? Why or why not? Note that:
    i. Matching must be specific e.g. '(' matches with ')' but not with '}' nor with ']' and so on.
    ii. Matched strings can be nested arbitrarily e.g. [(((({}))))]{{[][]}} is valid; but
    iii. Matchings cannot overlap i.e. ([)] is not valid.
d. Use your model in AE4 for designing a solution for data buffering at intermediate stages in a multi-hop network.
e. Design an ADT MAX_STACK which apart from the usual LIFO List operations includes a *max* operation that finds and returns the maximum element in the list.

Programming Experiments

6. Choose a data representation and implement the ADT CurrencyMap in DP1.
    a. Choose a space efficient data representation and implement the ADT CurrencyMap in DP1 to exploit the fact that the rate is symmetric for a direct exchange between any two currencies i.e. If 1 INR equals 3 RMB then 1 RMB is 1/3 INR.
    b. Choose a space efficient data representation and implement the ADT CurrencyMap in DP1 under the scenario that arbitrary conversions are not possible e.g. INR can be converted to RMB, USD, FF, SF, or DM etc. but cannot be converted to PKR, SLR, BLR, JPY, or DNR.
7. Measure the time and space actuals for each of the options in PE1 and plot them against the analytical results obtained in AE2.
8. Choose a data representation for the revised model in AE5 and implement the operations.
9. Choose a data representation for the ADT MAX_STACK in DP4.

a. Would your representation change if the *pop* operation is not permitted?
b. Would your representation change if popping the current maximum element requires popping all values in between the current maximum and previous maximum element?