IIT Mandi: Course CS 501 – Advanced Data Structures and Algorithms[1]

Assignment (1)

Lecturer: H. Schweppe
TA: Reena, Runa

Out: Wednesday 2013/2/27
In: Monday 2013/3/4
This set of assignments are NOT graded, just for finding out what you already know (or don't know)

1/1
a) Implement a complete **binary tree** of maximal hight h by means of an array.
In a complete binary tree, every node has exactly two childs, except (perhaps) for the leaf level. Leaf nodes are all on the left side of the tree (or the tree is full).
So the task is to map the binary tree structure to a linear array.
b) How many nodes does a binary tree of height h have at most? (Proof)
c) Implement the find (key) operations.
d) Discuss the pros and cons of this implementation. Could you conceive an application (like binary search tree) where the array implementation is to be preferred? Why? (~½ of a page)

2/1
Use **mathematical induction** to show that when n is an exact power of 2, the solution of the recurrence

$$T(n)=\begin{cases} 2 & \text{if } n=2 \\ 2T(\frac{n}{2}) & \text{if } n=2^k, k>1 \end{cases}$$

is

$$T(n)=n\log n$$

3/1

a) A sequence of $n > 0$ integers is called a *jolly jumper* if the absolute values of the differences between successive elements take on all possible values 1 through n - 1. For instance,

```
1 4 2 3
```

is a jolly jumper, because the absolute differences are 3, 2, and 1, respectively. The definition implies that any sequence of a single integer is a jolly jumper. Write a program to determine whether each of a number of sequences is a jolly jumper. The sequences we want to test, have length less or equal 100. (Python implementation requested).

b) if we allow for one million numbers in a sequence, discuss which data structure to use (from those you learned about in the 'Basic Data structures' course)

4/1
Prove the correctness of the following algorithm for evaluating a polynomial.
$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$

1 Please note: some of the exercises are from my previous course (many years ago...), many are from text books and other courses. I will not always cite the originator

```
function horner(A, x)
     p = A_n
     for i from n - 1 to 0
          p = p * x + A_i
     return p
```

5/1

A common but insecure method of encrypting text is to permute the letters of the alphabet. In other words, each letter of the alphabet is consistently replaced in the text by some other letter. To ensure that the encryption is reversible, no two letters are replaced by the same letter.

Your task is to decrypt several encoded lines of text, assuming that each line uses a different set of replacements, and that all words in the decrypted text are from a dictionary of known words.

Input

The input consists of a line containing an integer *n*, followed by *n* lowercase words, one per line, in alphabetical order. These *n* words compose the dictionary of words which may appear in the decrypted text. Following the dictionary are several lines of input. Each line is encrypted as described above.

There are no more than 1,000 words in the dictionary. No word exceeds 16 letters. The encrypted lines contain only lower case letters and spaces and do not exceed 80 characters in length.

Output

Decrypt each line and print it to standard output. If there are multiple solutions, any one will do. If there is no solution, replace every letter of the alphabet by an asterisk.

Sample Input

```
6
and
dick
jane
puff
spot
yertle
bjvg xsb hxsn xsb qymm xsb rqat xsb pnetfn
xxxx yyy zzzz www yyyy aaa bbbb ccc dddddd
```

Sample Output

```
dick and jane and puff and spot and yertle
**** *** **** *** **** *** **** *** ******
```

Programming assigments: use Python. Those who do not know Python can use C or Java, but ONLY THIS TIME.