

# Collection of programming interview questions

Željko Vrba

November 10, 2007

Most of this collection of problems has been taken from <http://www.gowrikumar.com/>. Additional problems will be added as I come across them. I have converted the problems to Latex, reformulated some of the text, and added some notes when the problem statement seemed vague (my notes are clearly marked with “[*Note*]”).

PROBLEM 1 Write an efficient C function `f` to count the number of bits set in an unsigned integer. For example: `f(0) = 0`, `f(1) = 1`, `f(2) = 1`, `f(3) = 2`.

PROBLEM 2 Write a small C program, which while compiling takes another program from input terminal, and on running gives the result for the second program. (*Hint*: Think UNIX.) Suppose the program is `1.c`; then compiling and executing the program should have the following effect:

```
# cc -o 1 1.c
int main()
{
    printf("Hello World\n");
}
^D
# ./1
Hello World
```

PROBLEM 3 Given strings `s1` and `s2`, write a snippet to say whether `s2` is a rotation of `s1` using only one call to the `strstr` function. For example, given `s1 = "ABCD"` and `s2 = "CDAB"`, return true; given `s1 = "ABCD"`, and `s2 = "ACBD"`, return false.

PROBLEM 4 What should be the `<condition>` so that the following code snippet prints "HelloWorld"?

```
if(<condition>)
    printf ("Hello");
else
    printf("World");
```

PROBLEM 5 Ugly numbers are numbers whose only prime factors are 2, 3 or 5. The sequence 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ... shows the first 11 ugly numbers. By convention, 1 is included. Write a program to find and print the 1500'th ugly number.

[*Note:* These numbers are also known as "Hamming numbers" or 5-smooth numbers.]

PROBLEM 6 Write a C program which when compiled and run, prints out a message indicating whether the compiler that it is compiled with allows `/* */` comments to nest.

PROBLEM 7 Write a C function that will take an int parameter  $n$  and print on stdout numbers 1 to  $n$ , one per line. The function *must not use* while, for, do-while loops, goto statement, recursion, or switch statement.

PROBLEM 8 Change/add only one character so that the following program prints \* exactly 20 times. (There are atleast 3 solutions to this problem.)

```
int main()
{
    int i, n = 20;
    for (i = 0; i < n; i--)
        printf("*");
    return 0;
}
```

PROBLEM 9 You are given have a datatype, say  $X$  in C. Determine the size of the datatype, without declaring a variable or a pointer variable of that type, and, of course without using the `sizeof` operator!

PROBLEM 10 Find the midpoint of a singly-linked list in a single pass through the list. Changing the list elements or links is not allowed.

PROBLEM 11 Given a *circular* singly-linked list of sufficiently large number of nodes (say more than 1), delete the node  $P$  given the pointer to it. Suggest the most efficient algorithm to delete the node, without going around the list.

PROBLEM 12 Write a C Program to reverse a stack in place using recursion. You can only use the following ADT functions on stack: `IsEmpty`, `IsFull`, `Push`, `Pop`, `Top`.

PROBLEM 13 You are provided with two stacks, as well as `Pop` and `Push` functions over them. Implement queue i.e. `Enqueue` and `Dequeue` using the available operations.

PROBLEM 14 Describe an algorithm for reversing the words in a string; for example "My name is Amit Agarwal" becomes "Agarwal Amit is name My". An  $O(n)$  and in-place solution is appreciable.

PROBLEM 15 You are given a sequence of numbers from 1 to  $n - 1$  with *one* of the numbers repeating only once; e.g. 1 2 3 3 4 5.

- How can you find the repeating number?

- What if given the constraint that you can use only a fixed amount of memory (i.e. independent of  $n$ )?
- What if there are two repeating numbers instead of one (with the same memory constraint)?

[*Note:* The problem is a bit vague. I think that the sequence should contain  $n$  numbers in range  $[1, n - 1]$ , and that it is *not* sorted in the general case (the example could be misleading).]

PROBLEM 16 In an integer array all numbers occur exactly twice, *except* for a single number which occurs exactly once. Give an  $O(n)$  algorithm to find the number occurring only once.

PROBLEM 17 There is a sequence of increasing numbers that have the same number of binary 1s in them. Given  $n$ , the number of 1 bits set in each number, write an algorithm or C program to find the  $n$ 'th number in the series.

PROBLEM 18 Given a stack  $S$ , write a C program to sort the stack (in ascending order). You are not allowed to make any assumptions about how the stack is implemented; the only functions allowed to be used are: `Push`, `Pop`, `Top`, `IsEmpty`, `IsFull`.

PROBLEM 19 Given a singly-linked list and an integer  $n$ , find the  $n$ 'th element from the end of the list.

PROBLEM 20 The numbers are represented with linked-list where each node represents a digit; for example:  $123 = 1 \rightarrow 2 \rightarrow 3 \rightarrow \emptyset$ ,  $999 = 9 \rightarrow 9 \rightarrow 9 \rightarrow \emptyset$  (where  $\emptyset$  denotes the NULL pointer).

Write a C program to add two numbers; e.g.  $9 \rightarrow 9 \rightarrow 9 \rightarrow \emptyset + 1 \rightarrow \emptyset = 1 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow \emptyset$

PROBLEM 21 There is a 100-story building and you are given two eggs. The eggs have an interesting property that if you throw the egg from a floor number  $< x$ , it will not break, and it will always break if the floor number is  $\geq x$ . Assuming that you can reuse the eggs which didn't broke, give an algorithm to find  $x$  in minimal number of throws.

PROBLEM 22 You are given a singly link-list such that each node of this list is also a head of another link list of the same type:

```
struct node {
    void *data; /* could be anything */
    struct node *next;
    struct node *down;
};
```

Describe an algorithm for flattening the list.

PROBLEM 23 Given two unsorted singly-linked lists, find the most efficient way to make a single sorted list.

PROBLEM 24 Given numbers  $x$  and  $n$ , where  $n$  is a power of 2, write (in C) a function  $f(x,n)$  which gives the [least] multiple of  $n$  that is greater than or equal to  $x$ . For example:  $f(13, 8) == 16$ ,  $f(17,16) == 32$ .

For “bonus points”: do not use division or modulo operator.

PROBLEM 25 In C, copying of array as follows is not possible:

```
int a[10],b[10];
a = b;
a = GetAnArrayOfTenElements();
```

Can you think of a simple hack, which would enable you to get this effect?

PROBLEM 26 Assume that two robots land from two different flights with the help of parachutes (at different points) on an infinite plane. Each robot leaves its parachute on landing and goes in search of the other robot. The problem is to write a program which will be executed by both the robots for rendezvous.

You have the following instructions at your disposal to write the program:

- Step L - makes the robot take one step towards left.
- Step R - makes the robot take one step towards right.
- Goto label - jump to the label.
- When P then [*one instruction*].

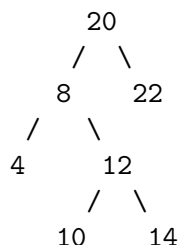
P is a predicate which will be true when a robot senses a parachute. It will be true as long as the robot is near it. The moment it takes a step towards left or right after sensing it, then P will become false. As part of 'if' statement one can execute one instruction which could be (1) or (2) or (3)

Write a program to help the robots to meet. Make your own assumptions and justify them. Remember that the same program needs to be executed by both robots.

PROBLEM 27 Given the values of two nodes in a binary search tree, write a C program to find the lowest common ancestor. You may assume that both values already exist in the tree. The function prototype is as follows:

```
int FindLowestCommonAncestor(node* root,int value1,int value)
```

For example, given the tree



and 4 and 14 as input, the function should return 8. Here the common ancestors of 4 and 14, are {8, 20}; 8 is the lowest of them.

PROBLEM 28 Given an array of  $2n$  elements in which  $n$  elements are same and the remaining  $n$  elements are all different, write a C program to find out the value of the  $n$  repeated elements. The elements appear in arbitrary order.

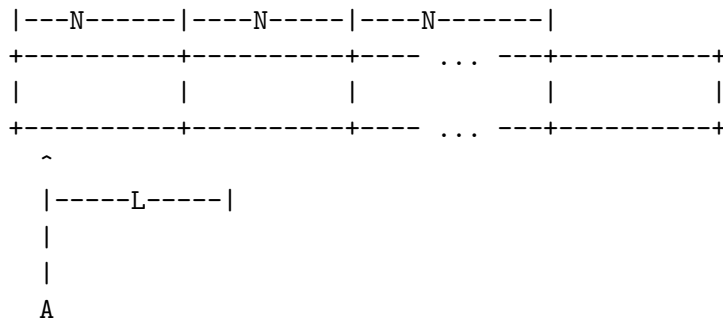
PROBLEM 29 You are given two arrays, A and B. Array A contains all elements of the array B, and *one extra element*. Find the value of the extra element.

Bonus: Solve the problem without using relational operators (<, >, etc).

PROBLEM 30 The distance between cities A and B is 1000 km. We have 3000 bananas in city A, and an elephant that can carry at most 1000 bananas at once. The elephant must eat a banana every 1km; it cannot go further if it runs out of the banana supply. What is the maximum number of bananas that can be transferred to the city B?

PROBLEM 31 Assume memory is divided into blocks starting at address 0 and of size  $N$  which is a power of 2. The blocks may be words, doublewords, pages, and so on. Given a starting address  $a$  and a length  $l$ , determine whether or not the address range from  $[a, a + l - 1]$  crosses a block boundary. The quantities  $a$  and  $l$  are unsigned and any values that fit in a register are possible.

Pictorially:



Write a C function which test whether  $a + l$  crosses a block boundary.

Bonus: do not use division or modulo operators.

PROBLEM 32 Write an efficient C function that deletes characters from a string. The function shall have the prototype

```
void RemoveChars(char str[], char remove[]);
```

where any character existing in `remove` must be deleted from `str`.

For example, given a `str` of "Battle of the Vowels:Hawaii" and `remove` of "aeiou", the function should transform `str` to "Bttl f th Vwls:Hw". Justify any design decisions you make and discuss the efficiency of your solution.

PROBLEM 33 Write an algorithm to check whether a given unsigned number is a multiple of 3, *without using division and modulo operators*.

PROBLEM 34 Prove that  $n(n + 1)(2n + 1)$  is divisible by 6, for any  $n > 0$ .

PROBLEM 35 Given an 8x8 chessboard, calculate:

- The number of subsquares in the board.
- The number of subrectangles in the board.

Note that rectangles are restricted to having different width and height.

PROBLEM 36 Describe the stepwise procedure for subtracting mixed fraction  $8\frac{5}{12} - 2\frac{2}{3}$ .  
*Note:* This is not a test of one's analytical skill, it's just to see how many people do it the easier way.

PROBLEM 37 Consider an array containing positive and negative integers. You have to find out the maximum a) sum, b) product possible by adding / multiplying  $n$  consecutive integers in the array, with  $n \leq$  array size. Also find the starting index of the sequence.

PROBLEM 38 Given two *sorted* linked lists,  $l_1$  and  $l_2$ , write a C program to compute their intersection  $l_1 \cap l_2$ .

PROBLEM 39 Propose a data structure that supports in  $O(1)$  time all of the following operations: Push, Pop, and FindMin which returns the smallest element present in the stack.

PROBLEM 40 A man has two cubes on his desk. Every day he arranges both cubes so that the front faces show the current day of the month. What numbers are on the faces of the cubes to allow this?

PROBLEM 41 Given 6 match-sticks of equal length, you are asked to form 4 equilateral triangles. You are not allowed to break, split or bend the sticks.

PROBLEM 42 Say we have a data structure as follows:

```
enum {RED,BLUE,GREEN};
struct Ball
{
    /*...*/
    int color;
};

int ColorOfBall(Ball b)
{
    return b.color;
}
Ball arr[SIZE];
```

The array `arr` consists of balls of with one of the three colours. Sort the array in such a way that all the red balls come first, followed by blue and then green. The restriction is that call to function `ColorOfBall` very costly, and it has to be used as little as possible. In other words, find a solution with the least number of calls to the function `ColorOfBall`.

PROBLEM 43 Propose a data structure that holds elements from 0 to  $n - 1$  and supports *all* of the following operations in  $O(1)$  time: initialization, insertion of an element, deletion of an element, finding an element, deleting all elements.

[*Note*: I seem to remember seeing this as an exercise in one of the Knuth's books.]

PROBLEM 44 Given 13 balls, how would you arrange them in 9 lines, such that there are 4 balls in each line? You can assume that the lines are arranged in 2-D space and that a ball cannot be placed on top of another ball.

Bonus: if you find that too easy, and have loads of time to kill, then how about arranging 22 balls in 21 lines of 4?

PROBLEM 45 Write a C function

```
int AddOvf(int* result, int a, int b)
```

which returns 0 and places the sum  $a+b$  into  $*result$  if the addition can be performed without overflow. Otherwise (in the case of overflow), the function returns -1. You are *not* allowed to cast integers to long to check for the overflow.

PROBLEM 46 Write a C function that rotates right an unsigned integer by  $n$  bits.

PROBLEM 47 A skier must decide every day she goes skiing whether to rent or buy skis, unless or until she decides to buy them. The skier does not know how many days she will go on skiing before she gets tired of this hobby. Suggest a strategy for the skier minimizing her cost, given the cost of rent is 1 unit, and the cost to buy the skis is  $B$  units where  $B \gg 1$ .

PROBLEM 48 Can you explain the behaviour of the following C program?

```
int main()
{
    float f=0.0f;
    int i;

    for(i=0;i<10;i++)
        f += 0.1f;
    if(f==1.0f)
        printf("f is 1.0f\n");
    else
        printf("f is NOT 1.0f\n");

    return 0;
}
```

PROBLEM 49 You need to write a simple macro `CHAR(x)` which takes a character  $x$  and converts it into ASCII value of  $x$ :

```
printf("%c",CHAR(a)) ==> a
printf("%c",CHAR(X)) ==> X
```

PROBLEM 50 Here is a small one to drive away the afternoon siesta: consider a point  $P(n, n)$  in the cartesian co-ordinate system. A robot has to start from the origin and reach this point. The only steps the robot can take are 1 unit right or 1 unit up. How many different paths can the robot take to point  $P$ ? Is there an optimal path to point  $P$  (both up and right steps incur the same cost)?

PROBLEM 51 How could you prevent a class from being used as a base class in C++? For example: consider some class ABC. I would like the user to create objects of class ABC but prevent him from deriving classes from ABC and creating objects of the derived class.

PROBLEM 52 A multiple choice test has 15 questions and 4 choices for each answer. Each question has only one answer correct. In how many ways can the 15 questions be answered so that:

- Exactly 3 answers are correct
- Atleast 3 answers are correct

PROBLEM 53 What does the following chunk of code do?

```
cwd
xor ax, dx ; ax := ax xor dx
sub ax, dx ; ax := ax - dx
```

cwd converts a 16-bit word into sign-extended 32-bit double word stored in dx:ax registers.

PROBLEM 54 What is the smallest / largest possible number of friday 13ths in a year?

PROBLEM 55 Write a recursive function which generates the power set of a given set. The set is passed to the function as a string, and the function should print the subsets as strings.

Example: given "abc", the function should print  $\epsilon$  (the empty string, encode it somehow), a, b, ab, c, ac, bc, abc.

PROBLEM 56 Bangla numbers normally use 'kuti' (1000000), 'lakh' (100000), 'hajar' (1000), 'shata' (100) while expanding and converting to text. You are going to write a program to convert a given number to text with them.

The input file may contain several test cases. Each case will contain a non-negative number  $\leq 999999999999999$ . For each input case, you have to output a line starting with the case number with four digits adjustment, followed by the converted text.

Sample Input

```
-----
23764
45897458973958
```

Sample Output

```
-----
1. 23 hajar 7 shata 64
2. 45 lakh 89 hajar 7 shata 45 kuti 89 lakh 73 hajar 9 shata 58
```

Note: this problem is at <http://acm.uva.es/p/v101/10101.html>.

PROBLEM 57 Suppose that there are 101 players entered in a "single elimination" tennis tournament. In such a tournament any player who loses a match must drop out, and every match ends in a victory for some player - there are no ties. In each round of the tournament, the players remaining are matched into as many pairs as possible, but if there is an odd number of players left, someone receives a bye (which means an automatic victory for this player in this round). Enough rounds are played until a single player remains who wins the tournament. How many matches are played in total?

PROBLEM 58 The "parity" of a number refers to whether it contains an odd or even number of 1-bits. The number has "odd parity", if it contains odd number of 1-bits and is "even parity", if it contains even number of 1-bits. Write a C function to find the parity of an unsigned integer.

PROBLEM 59 Calculate:  $a^b \bmod m$  for large values of  $a$ ,  $b$  and  $m$  using an efficient algorithm. Value ranges:  $a \in [0, 2147483647]$ ,  $b \in [2147483647]$ ,  $m \in [1, 46340]$ .

PROBLEM 60 Lets assume we have a rat maze as represented by the following  $n \times m$  matrix where S is the start location and F is the end location.

S	0	0	0	0	0
1	1	0	0	0	0
1	0	1	0	0	0
1	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
0	1	1	1	1	F

The idea (as with any rat maze) is to traverse from S to F. The matrix can have only 0 and 1 as values. 1 represents a path that can be taken and 0 represents a blocked path. We can make the following assumption: S will always be (0,0) and F will always be (n,m).

How to find the shortest (or longest) path from S to F without actually traversing all the possible paths?

PROBLEM 61 If two of the below statements are false, what chance is there that the egg came first?

1. The chicken came first.
2. The egg came first.
3. Statement I is false and Statement II is true.

PROBLEM 62 The following program, when run,

```
#include <stdio.h>
main()
{
```

```

int a,b,c;
int count = 1;
for (b=c=10;a="- FIGURE?, UMKC,XYZHello Folks,\
TFy!QJu R0o TNn(ROo)SLq SLq ULo+\
UHs UJq TNn*RPn/QPbEWS_JSWQAIJO^\
NBELPeHBFHT}TnALVlBLOFAkHF0uFETp\
HCStHAUFAgcEAelclcn^r^r\\tZvYxXy\
T|S^Pn SPm SOn TNn ULoOUlo#ULo-W\
Hq!WFs XDt!" [b+++21]; )
for(; a-- > 64 ; )
putchar ( ++c=='Z' ? c = c/ 9:33^b&1);
return 0;
}

```

gives the following output:



Can you figure out the logic used in the program?

PROBLEM 63 What are the next few numbers in the sequence 1, 20, 33, 400, 505, 660, 777, 8000, 9009, 10100, 11121?

PROBLEM 64 You are travelling in the jungles of Africa, when you are caught by a tribe of barbarians. They allow you to choose between death or solving their great challenge:

You are blindfolded and taken to a room, where you are asked to kneel. You feel hundreds of circular discs lying in front of you. You are told that one side of each disc is painted red, and the other, green. There are exactly 129 discs that currently are red side up. You have to divide the discs into two groups, such that each group has the same number of discs showing the red colour. Obviously, no peeking allowed.

PROBLEM 65 Given an array of strings, you need to sort it using "qsort" function provided by C. You may use the `strcmp` function provided by the C library.

```
#define SIZEOF(arr) (sizeof(arr)/sizeof(arr[0]))
int main()
{
    char *arr[]{"abc","def","abcd"};
    int i;

    /*
     * code to sort..
     *...
     *...
     */

    for(i=0;i < SIZEOF(arr);i++)
        printf("%s\n",arr[i]);
}
```

PROBLEM 66 Here is a game one can buy in most toy stores, it's called Hi-Q (or Brainvita). 32 pieces are arranged on a board as shown below:

```

      X X X
      X X X
X  X X X X X X
X  X X 0 X X X
X  X X X X X X
      X X X
      X X X
```

The rules are as follows:

- In the beginning, only the centre position is unoccupied.
- A piece is allowed to move by jumping over one of it's neighbours into an empty space.
- Diagonal jumps are not permitted.
- When a piece is jumped over, it is removed from the board.

Write an algorithm which determines a series of jumps so that all of the pieces except one are eventually removed, and the final piece ends up at the center position.

PROBLEM 67 Suppose we wish to multiply four matrices of real numbers  $M_1 \times M_2 \times M_3 \times M_4$  where  $M_1$  is 10 by 20,  $M_2$  is 20 by 50,  $M_3$  is 50 by 1, and  $M_4$  is 1 by 100. Assume that the multiplication of a  $p \times q$  matrix by a  $q \times r$  matrix requires  $pqr$  scalar operations, as it does in the usual matrix multiplication algorithm.

Find the optimal order in which to multiply the matrices so as to minimize the total number of scalar operations. How would you find this optimal ordering if there are an arbitrary number of matrices?

PROBLEM 68 Describe a  $O(n \lg n)$  time algorithm that, given a set  $S$  of  $n$  real numbers and another real number  $x$ , determines whether or not there exist two elements in  $S$  whose sum is exactly  $x$ .

PROBLEM 69 Given an array of  $n$  numbers  $a_1 \dots a_n$ , find the second minimum number in  $n + \lg n$  comparisons. You can only compare elements and you can't assume anything about the range of values of the numbers.

PROBLEM 70 An element in a sorted array can be found in  $O(\lg n)$  time via binary search. But suppose I rotate the sorted array at some pivot unknown to you beforehand. So for instance, 1 2 3 4 5 might become 3 4 5 1 2. Now devise a way to find an element in the rotated array in  $O(\lg n)$  time.

PROBLEM 71 Write a program that will display a "spiral" of  $n \times n$  numbers, using *constant space* (no arrays allowed). For example, here's what the spiral looks like for  $n = 10$ :

99	98	97	96	95	94	93	92	91	90
64	63	62	61	60	59	58	57	56	89
65	36	35	34	33	32	31	30	55	88
66	37	16	15	14	13	12	29	54	87
67	38	17	4	3	2	11	28	53	86
68	39	18	5	0	1	10	27	52	85
69	40	19	6	7	8	9	26	51	84
70	41	20	21	22	23	24	25	50	83
71	42	43	44	45	46	47	48	49	82
72	73	74	75	76	77	78	79	80	81

PROBLEM 72 How do you efficiently find the largest repeating string and the number of times it repeats in a given string? For example, given string "abc fghi bc kl abcd lkm abcdefg", the function should return string "abcd" and the count of 2.

PROBLEM 73 There are 4 buckets of 9 coins each. Real coins weigh one gram each, and fake coins weigh 2 grams each. Each bucket is either fake (contains only fake coins) or real (contains only real coins). Given a weighing machine, how can you determine all the buckets that are fake with just one weighing?

PROBLEM 74 Given a binary tree, how do you verify whether it is a binary search tree?

PROBLEM 75 Write a C function, which takes a number  $n$  and positions  $p_1$  and  $p_2$  and returns whether the the bits at positions  $p_1$  and  $p_2$  are the same or not.

PROBLEM 76 Write a C function, which takes two numbers  $m$  and  $n$  (representing sets of elements;  $i$ 'th bit is 1 if  $i$  is an element of the set), and checks whether  $m$  is a subset of  $n$  or not.

PROBLEM 77 A majority element in an array  $A$ , of size  $n$  is an element that appears more than  $n/2$  times (and hence there is at most one such element) Write a function which takes an array and emits the majority element (if it exists), otherwise prints NONE. Bonus: solve the problem in  $O(n)$  time.

For example, the majority element in 3 3 4 2 4 4 2 4 4 is 4, while the sequence 3 3 4 2 4 4 2 4 has no majority element.