Appendix: Quiz on Individual Topics

Question 1. A hacker wants to use *Arithmetic coding* that would send a virus to the client's computer and unpack itself there. His favorite virus uses this RNS sequence: `GACGU$`, where `A`, `C`, `G`, `U` are four nucleobases (the useful data payload), but the symbol `$` is used just once to mark the end of the RNS string.
He uses the following *a priori* frequencies for the symbols:

A	C	G	U	$
30%	10%	30%	20%	10%

He starts with the half-closed line segment $S_0 = [0; 1)$ and at every step (for all $i = 0, 1, 2, 3, 4, 5$) creates the next segment S_{i+1} from S_i by dividing S_i into five parts of lengths proportional to the frequencies (the proportions of subdivisions are $3:1:3:2:1$). Then S_{i+1} is the subdivision of the previous S_i corresponding to the newly encoded character. After encoding all six characters in the virus message `GACGU$`, the hacker gets the segment S_6.

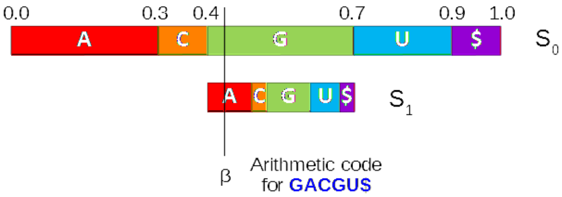


Figure 1. Building arithmetic code

Find the binary fraction β belonging to S_6.
Select one answer.
(A) $\beta = 0.011011101100011_2$
(B) $\beta = 0.011011101100110_2$
(C) $\beta = 0.011011101101001_2$
(D) $\beta = 0.011011101101100_2$
(E) $\beta = 0.011011101101111_2$

Note. In arithmetic coding any sequence of n bits represents an interval of length $\frac{1}{2^n}$. For example the bit sequence `010` stands for the segment of real numbers: $[0.010_2; 0.011_2) = [0.25; 0.375)$ having length $\frac{1}{2^3} = \frac{1}{8}$. If the arithmetic coding yields a segment $[a; b)$ such that $[0.25; 0.375) \subseteq [a; b)$, then `010` is the result of the arithmetic coding.
The number $\beta = 0.010_2 \in [a; b)$, and the extra "0" character ensures that the interval is not too long and it fits inside $[a; b)$. For example, `01` would represent a different interval $[0.25; 0.5) \neq [0.25; 0.375)$.

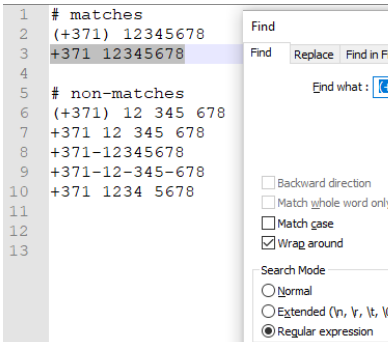
Question 2. We want to use a *regular expression* to find all phone numbers with the Latvian country code. Assume that the phones can have one of the following formats (here symbol `D` denotes any digit (0-9)). The space symbols are used exactly as written (they are single space characters).

```
(+371) DDDDDDDD
+371 DDDDDDDD
```

Pick a regular expression that would recognize just these 2 phone formats.
Select one answer.

```
(A)   (+371|\(+371\)) \d{8}
(B)   (\+371|\(\+371\)) [0-9]{8}
(C)   (+371|\(+371\)) \d\d\d\d\d\d\d\d
(D)   \+371|(\+371\) [0-9]{8,8}
```

Note. If you wish, you can use text editor such as Notepad++ search dialogue to verify which regex works. (Click **Ctrl+F**, enter your regular expression, switch "Search mode" to Regular Expression, and click the button "Find All in All Opened Documents"):



```
# matches
(+371) 12345678
+371 12345678

# non-matches
(+371) 12 345 678
+371 12 345 678
+371-12345678
+371-12-345-678
+371 1234 5678
```

Find what :
☐ Backward direction
☐ Match whole word only
☐ Match case
☑ Wrap around
Search Mode
○ Normal
○ Extended (\n, \r, \t, \
● Regular expression

Figure 2. Regex Search in Notepad++

If you are on a Linux machine, you can also test regex search with the command "grep" (and flag -E).

Question 3. About 70% of the entries in a bit array of a *Bloom filter* are equal to 1 as it is initialized with the words from some dictionary D. The Bloom filter computes $n = 8$ independent hash functions to check, if some entry belongs to the dictionary D. What is an (approximate) chance P to get a false positive? A false positive is an event where one picks a random word $w \notin D$, and Bloom filter incorrectly states that w belongs to D

Select one answer.
(A) $P = 30.00\%$,
(B) $P = 5.76\%$,
(C) $P = 3.75\%$,
(D) $P = 0.66\%$.

Question 4. A testing laboratory tests the same number of people daily, and on day i, the number of people who tested positively for some health condition was n_i. The laboratory knows that the numbers n_i are distributed according to *Poisson distribution* with the expected value $\lambda = 13.5$.
By P we denote the probability that on the given day

$n_i < 5$ (i.e. less than 5 people test positive for that condition). Which is the closest approximate value for this probability?
(A) $P = 0.26\%$,
(B) $P = 0.51\%$,
(C) $P = 5.78\%$,
(D) $P = 10.89\%$.

Note. Another typical illustration of the same Poisson distribution: Imagine the ice cream "Rūjienas saldējums" with raisins. (In this case the average number of raisins in one package is $\lambda = 13.5$; and you have to find the probability that a given ice cream package has at most 4 raisins. See https://bit.ly/2KanwIf.

Figure 3. An Ice Cream Package with Raisins satisfying Poisson Distribution

Question 5. Find the minimum number of colors to paint the 12 vertices from the graph shown in Figure 4 so that any two vertices connected with an edge are having different colors. (This number n is called the *chromatic number* for the graph G.)

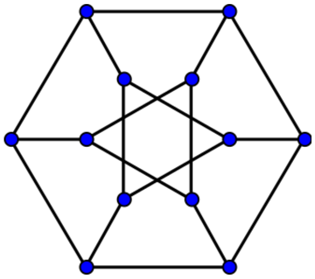


Figure 4. Graph for Vertex Coloring

Question 6. Assume that the "World Wide Web" contains only 4 pages A, B, C, D that link to each other as shown in the picture below.

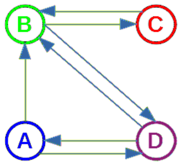


Figure 5. Four webpages with links.

In the Iteration 0 initialize the page ranks with equal values

$$\mathrm{PR}_0(A) = \ldots = \mathrm{PR}_0(D) = \frac{1}{4}.$$

Compute the first two iterations of these pageranks, using the formulas:

$$\begin{cases} \mathrm{PR}_{i+1}(A) = (1-d) + d\left(\frac{\mathrm{PR}_i(D)}{2}\right) \\ \mathrm{PR}_{i+1}(B) = (1-d) + d\left(\frac{\mathrm{PR}_i(A)}{2} + \frac{\mathrm{PR}_i(C)}{1} + \frac{\mathrm{PR}_i(D)}{2}\right) \\ \mathrm{PR}_{i+1}(C) = (1-d) + d\left(\frac{\mathrm{PR}_i(B)}{2}\right) \\ \mathrm{PR}_{i+1}(D) = (1-d) + d\left(\frac{\mathrm{PR}_i(A)}{2} + \frac{\mathrm{PR}_i(B)}{2}\right) \end{cases}$$

Set the value of the damping factor $d = 0$.
Write the values of the second iteration for all the pages: $\mathrm{PR}_2(A), \ldots, \mathrm{PR}_2(D)$..
Write 4 comma-separated numbers; round them to the nearest thousandth.

Note. You can also use vector algebra, if you are familiar with multiplying matrices with vectors – https://bit.ly/2RJTNtC.

$$\begin{pmatrix} \mathrm{PR}_2(A) \\ \mathrm{PR}_2(B) \\ \mathrm{PR}_2(C) \\ \mathrm{PR}_2(D) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix}^2 \cdot \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}.$$

In this formula a square matrix 4×4 is twice multiplied to a 4×1 vector $(1/4, 1/4, 1/4, 1/4)$ from the left side.

Note. https://checkpagerank.net/check-page-rank.php shows that:
`https://www.bitl.lv/` has PageRank 2/10,
`https://www.delfi.lv/` has PageRank 6/10.
This does **not** mean that there are three times more inbound links to Delfi than to BITL (or that these links are three times more "valuable"). The value returned by this web resource is a *logarithmic measure* of its iterative value. In fact, the difference between 2/10 and 6/10 means that the popularity of these pages differs by many orders of magnitude. See https://bit.ly/2yrRMf7.

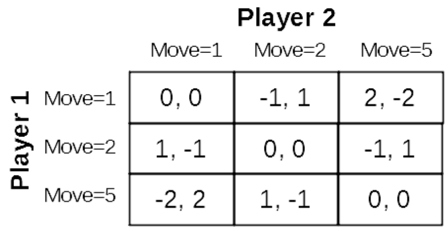
Question 7. As you probably know, *Karatsuba's algorithm* can express the multiplication of two numbers of length n digits as three multiplications of numbers of length $n/2$ (i.e. the number of operations are three times larger, but the operands become two times shorter). This ultimately means that Karatsuba's algorithm requires only $O(n^{1.585})$ operations to multiply numbers of length n.
Imagine that somebody has invented a new operation $a \otimes b$ for some objects a, b (both a, b have the same size n). Assume that s/he knows how to express $a \otimes b$ using 7 operations $a_i \otimes b_i$ (where $i = 1, 2, \ldots, 7$, and all a_i, b_i have size $n/2$, i.e. half the size of the original operands a, b). (*We do not care, what the operation $\otimes$ does; but we know that we can compute it for arguments a, b of*

length 1 *in constant time; it is therefore easy for very short arguments.*)
Find the best Big-O-Notation estimate for the time needed to compute $a \otimes b$, if a, b are both of size n.
(A) $O(n^2)$
(B) $O(n^2 \log n)$
(C) $O(n^{2.646})$
(D) $O(n^{2.808})$
(E) $O(n^3)$
(F) $O(n^3 \log n)$
Select one answer.

Note 1. One can use Master's theorem (Rosen2019, p.558) for this problem and also for the Karatsuba's algorithm.

Note 2. If the algorithm falls in multiple Big-O complexity classes, pick the one that shows the slowest growth. For example, if a speed of an algorithm is both in $O(n^2)$ and $O(n^3)$, then $O(n^2)$ would be a more precise and a more useful estimate.

Question 8. Assume that two players A and B play a *matrix game*. They simultaneously guess one number each. Either player can guess one of these three numbers: $\{1, 2, 5\}$. The payoff matrix is shown below. In each cell the first number is what is paid to player A, the second number is paid to player B.



		Player 2		
		Move=1	Move=2	Move=5
Player 1	Move=1	0, 0	-1, 1	2, -2
	Move=2	1, -1	0, 0	-1, 1
	Move=5	-2, 2	1, -1	0, 0

Figure 6. Matrix game with payoffs.

Expressed in human language, the rules are as follows. Assume that the player A just guessed a number a, and player B guessed a number b.

- If $a = b$, then it is a tie; nobody pays anything.
- If $a > b$ (yet $a < 3b$), then B pays to A one euro. (And also, if $b > a$ yet $b < 3a$, then A pays to B one euro.)
- If $a \geq 3b$, then A pays to B two euros. (And also, if $b \geq 3a$, then B pays to A two euros.)

In this number guessing game one can win by guessing a number which is a little bit larger than the other player's number. But one should not guess a number which is larger than the other by "a lot" (if you exceed the other player's number three times or more, then you suffer a double loss.).
Which can be *Nash equilibrium* for this number guessing game? (You can assume that one of the answer variants is correct – the same optimal strategy for both players. It is enough to find the one that beats all the other strategies.) Each strategy lists the probabilities for guessing the number x:
(A) $P(x = 1) = 1/3$, $P(x = 2) = 1/3$, $P(x = 5) = 1/3$.
(B) $P(x = 1) = 0$, $P(x = 2) = 1/2$, $P(x = 5) = 1/2$.
(C) $P(x = 1) = 1/2$, $P(x = 2) = 1/2$, $P(x = 5) = 0$.
(D) $P(x = 1) = 1/4$, $P(x = 2) = 1/2$, $P(x = 5) = 1/4$.
(E) $P(x = 1) = 2/6$, $P(x = 2) = 3/6$, $P(x = 5) = 1/6$.
Select one answer.

Question 9. The first iterations using Lindermayer system are given:
Iteration 0: A
Iteration 1: AB
Iteration 2: ABBA
Iteration 3: ABBABAAB
Iteration 4: ABBABAABBAABABBA
(*To get Iteration 5: Take Iteration 4, change all A's into B's and vice versa, and append such string to the end of Iteration 4.*) Find the correct set of rules to generate this L-system.

(A) $\begin{cases} \mathtt{A} \to \mathtt{B} \\ \mathtt{B} \to \mathtt{BA} \end{cases}$

(B) $\begin{cases} \mathtt{A} \to \mathtt{AB} \\ \mathtt{B} \to \mathtt{AA} \end{cases}$

(C) $\begin{cases} \mathtt{A} \to \mathtt{AB} \\ \mathtt{B} \to \mathtt{BA} \end{cases}$

(D) $\begin{cases} \mathtt{A} \to \mathtt{AB} \\ \mathtt{B} \to \mathtt{AB} \end{cases}$

Select one answer.
Note. We can have a *turtle* that reads this sequence and performs actions:

- Letter A: Step 1 unit ahead, turn 60° counterclockwise.
- Letter B: Turn 180°.

In this case the iterations $0, 2, 4, \ldots$ would produce a fragment of Koch snowflake (Figure 7).

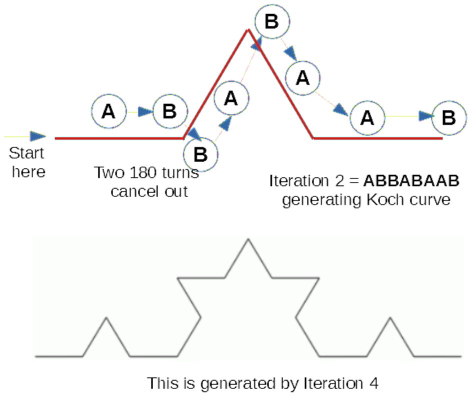

B
A
A
B
A
A
B
Start here
B
Two 180 turns cancel out
Iteration 2 = ABBABAAB generating Koch curve
This is generated by Iteration 4

Figure 7. Koch curve as an L-system

Question 10. Assume that someone uses a *secure hash* algorithm $h(x)$ that for any file x outputs a hash value consisting of exactly 100 bits. (The typical SHA-256 algorithm would return 256 bits.)

Assume that we want to use brute force to find hash collision – two different files x_1, x_2 such that $h(x_1) = h(x_2)$. You can estimate, how many hash values we need to compute before we get at least 50% probability to find a hash collision. Estimation can be done using Square aproximation from the Birthday paradox:

$$p_{\text{collision}} \approx \frac{n^2}{2m}, \qquad (1)$$

Formally: If a hash function $h(x)$ can take m different values and we randomly pick n different integer numbers $x_1, \ldots, x_n$, then the probability that there is at least one collision ($h(x_i) = h(x_j)$ and $x_i \neq x_j$) is approximately expressed by the formula (1). See https://bit.ly/2RNjhGB.

Assume that a single hash value $h(x)$ can be computed in one microsecond ($1\,\mu s = 10^{-6}\,s$). Estimate the number of years it would take to produce a collision for a 100-bit secure hash algorithm with probability at least 50%.

(A) The expected time is 0.11 years.
(B) The expected time is 35.7 years.
(C) The expected time is 71.4 years.
(D) The expected time is 856 years.
(E) The expected time is $4.02 \cdot 10^{16}$ years.

Select one answer.

Question 11. Consider the following problem solving strategies:

(A) Drawing a picture. Can you write down all the things you need to consider on paper? Can you order them nicely in a list or a table? Can you show them in a two-dimensional or a three-dimensional drawing?

(B) Getting hands dirty. Can you start experimenting with the problem, plug in specific values, see where they lead you?

(C) Going to the extremes. Can you pick some "borderline case"? Is there the smallest or the largest item that is possible in the problem?

(D) Lateral thinking. Could it happen that your current solving approach is not applicable or is too inefficient? Can you pretend that you have not spent many years studying mathematics at school; can you apply lateral/divergent thinking out of the box to come up with something unexpected?

(E) Looking for symmetries. Can we switch two numbers or two letters in our notation? Can we inspect just one item and notice that many others are identical?

(F) Making it easier. Can we make a simpler version of this problem and solve it first? Insert a smaller number? Solve only one particular case of it?

(G) Penultimate step. What precondition must take place before the final solution step is possible? Imagine, which result you would need in order to say that you are "almost done".

(H) Wishful thinking. Can you apply some outrageous simplification to your initial problem. Imagine for a while that you have already solved it: What would that imply?

Now consider the following problem:

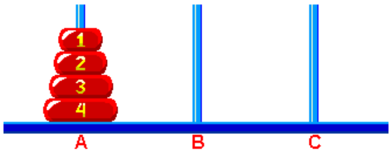


Figure 8. Tower of Hanoi

Problem. A Tower of Hanoi (Figure 8) has three pegs (A, B, C) and four disks initially on the peg A. The task is to move all the four disks to the peg B, where the following rules apply:
Rule 1: Only one disk can be moved at a time.
Rule 2: Each move consists of taking the upper disk from any peg and moving it to another peg.
Rule 3: No larger disk may be placed on top of a smaller disk.
The solver wants to come up with the sequence of moves. S/he has tried a similar game with just three disks with some trial and error, but is not sure how to proceed in the case with four disks. Somebody suggests a few "natural looking" hints.
Hint 1. Find the disk that is the hardest to move anywhere or moved least frequently?
Hint 2. To which peg all the other disks need to go before we move this disk?
Hint 3. Assume that you know how to move three disks from the peg A to the peg B. Can you move them between any other pegs? How?

What kind of problem solving strategies are contained in the hints?

Select up to three relevant strategies (A-H).

"Never, ever, think outside the box."

© *Leo Cullum*, https://www.newyorker.com/

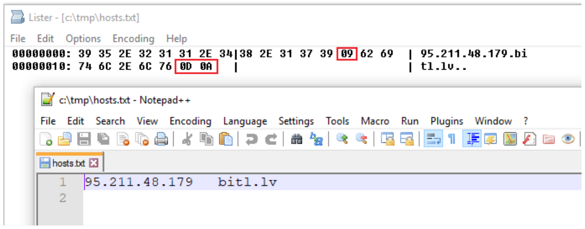
Question 12. Someone wants to compute a MD5 checksum for the following file:

```
95.211.48.179	bitl.lv
```

The following is true:

- File `hosts.txt` is exactly 23 bytes long.
- The IP address is seperated from `bitl.lv` by a single horizontal tab (byte in hexadecimal: `0x09`).
- The only line is ends with a Windows-style line ending (carriage return, line feed: bytes in hexadecimal: `0x0D`, `0x0A`).

Figure 9 shows file displayed by *Total Commander*; button **F3**, then menu item **Options** > **Hex** (and also in the Notepad++ editor).


Lister - [c:\tmp\hosts.txt]
File Edit Options Encoding Help

```
00000000: 39 35 2E 32 31 31 2E 34|38 2E 31 37 39 09 62 69  | 95.211.48.179.bi
00000010: 74 6C 2E 6C 76 0D 0A   |                         | tl.lv..
```

c:\tmp\hosts.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
hosts.txt

```
1  95.211.48.179	bitl.lv
2
```

Figure 9. Bytes in file `hosts.txt`

Copy the whole MD5 checksum in your answer.

Note. In this exercise it is important to have exactly the same file content as shown in the picture. For example, replacing the **TAB** character by one or more spaces (or Windows-style line ending with a UNIX-style line ending) would totally change MD5. (For secure hashes there is absolutely no string tokenization – unlike plagiarism detection they are very sensitive against the smallest changes in their input.)

Question 13. There are two people playing a game: Player A (he is the Maximizer – wants to go down to a leaf with maximum payoff), and Player B (he is the Minimizer – wants to minimize the A's payoff). The current position is the root of the tree (Figure 10) and it is Player's A turn to make the first move (to any of the root's children). After that Player B moves (going down one more level) and so on – until they reach a leaf, which shows the payoff for Player A.

Find the maximum payoff for Player A (you could use minimax algorithm with or without Alpha-Beta pruning speedup to find out).

Write the payoff as a positive integer.

Question 14. If you verify the Conway's game of life the configuration P_0 of a straight line with 4 live cells (Figure 11), you need $N = 2$ steps until you reach "periodic state" P_2 that will return after period $T = 1$ (i.e. returning needs just one step $P_3 = P_2$, since "beehive" configuration is stable). So in this case $(N, T) = (2, 1)$ – there are $N = 2$ preliminary steps, and after that there is a period of length $T = 1$.

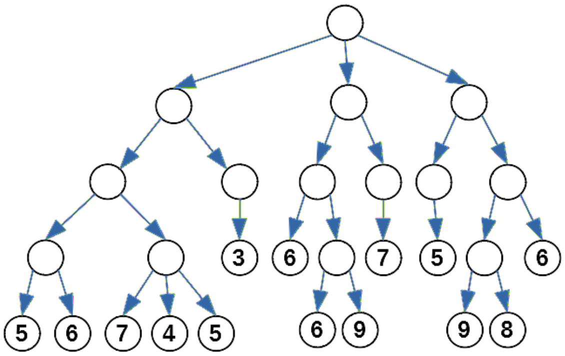


Figure 10. Game positions in a tree.

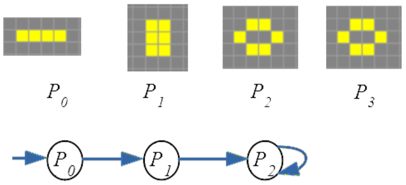


Figure 11. Conway game for a line of 4.

Now consider a different starting position P_0 that contains a straight line of 5 live cells (Figure 12). Determine the number of steps N needed to reach the first position P_N that would repeat infinitely often, and the length of period T with which the subsequent steps repeat, i.e. the smallest positive integer T with the property:

$$\forall k \in \mathbb{Z}_{0+},\ (k \geq N) \ \rightarrow\ (P_{k+T} = P_k).$$

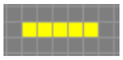


Figure 12. Starting position P_0 for a line of 5.

Write two comma-separated integers N, T.

Note. In Conway's game there are some positions that are not periodic (glider guns that constantly create new stuff), but most simple positions eventually reach periodic state. Therefore the numbers N and T are defined in these cases.

Question 15. A mathematical theory $\mathcal{T}$ (in a similar way as Coq software) provides rules to prove various mathematical statements. Assume that in this theory $\mathcal{T}$ one can prove some statement A and also the statement $\neg A$. Which description is true for this theory:

(A) $\mathcal{T}$ is consistent.
(B) $\mathcal{T}$ is not consistent.
(C) $\mathcal{T}$ is complete.
(D) $\mathcal{T}$ is not complete.
(E) $\mathcal{T}$ is effectively axiomatized.
(F) $\mathcal{T}$ is not effectively axiomatized.

Select one answer.

Question 16. Some banks can issue 19-digit credit card numbers (instead of the more typical 16-digit ones). Assume that there is a 19-digit number that satisfies the Luhn check (mod 10):

$$\texttt{557367054456450571}*.$$

Please find the digit that is written in the place of the last $*$ symbol.
Write a single digit.

Question 17. Some text T has been tokenized into a sequence of N words $(w_0, \ldots, w_{N-1})$. Assume that you assigned unique numbers to the stemmed words (each word w_i in the text T is replaced by a number $n(w_i)$); and then computed rolling hash values for five consecutive words in this text using this formula:

$$H(w_1, w_2, w_3, w_4, w_5) =$$
$$= \left(n(w_1) \cdot a^4 + n(w_2) \cdot a^3 + n(w_3) \cdot a^2 + n(w_4) \cdot a^1 + n(w_5)\right) \bmod q.$$

This is a polynomial value for the argument a followed by a remainder when dividing by q. Parameters a and q are two large primes.
We compute all such hash values:

$$\begin{cases} v_0 = H(w_0, w_1, w_2, w_3, w_4), \\ v_1 = H(w_1, w_2, w_3, w_4, w_5), \\ \ldots \\ v_{N-5} = H(w_{N-5}, w_{N-4}, w_{N-3}, w_{N-2}, w_{N-1}). \end{cases}$$

It turned out that 10% of these hash values were found in an existing hashtable H (built from some existing texts using the same hash function) – about 5% of the values in that hashtable are marked (the others are empty). What is the most likely explanation for this?
(A) Text T contains large chunks of text that is copy-pasted from other sources.
(B) Overlaps of the size 10% can happen by chance. On the other hand, overlaps exceeding $1/5$ (the size of the rolling hash window) would be highly unusual and would require manual inspection.
(C) This is not an effective way to detect copying and plagiarism. Rolling hash should instead run on characters (not entire words), since multiple authors may use the same words.
Select one answer.

Question 18. Jane took an ordinary soccer ball made from an elastic material (Figure 13).
She stretched one of its faces so that it became a planar graph (Figure 14). Then she marked a Hamiltonian cycle in this graph (not shown).
How many edges of this graph do **not** belong to the Hamiltonian cycle?
Write a positive number.

Figure 13. 3D Soccer Ball

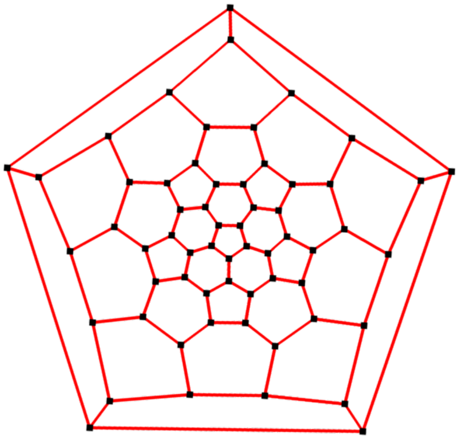


Figure 14. Planar Soccer Ball

Answers

Question 1. Answer **(D)**
The intervals encoding the string "GACGU$" form the following sequence:

$$\begin{cases} S_0 = [0.000000; 1.000000] \text{ encodes "" (empty string),} \\ S_1 = [0.400000; 0.700000] \text{ encodes "G",} \\ S_2 = [0.400000; 0.490000] \text{ encodes "GA",} \\ S_3 = [0.427000; 0.436000] \text{ encodes "GAC",} \\ S_4 = [0.430600; 0.433300] \text{ encodes "GACG",} \\ S_5 = [0.432490; 0.433030] \text{ encodes "GACGU",} \\ S_6 = [0.432976; 0.433030] \text{ encodes "GACGU\$".} \end{cases}$$

To make the sequence of intervals $S_0 \supset S_1 \supset \ldots \supset S_6$, we can define iterative sequences Left_i, Length_i describing the left endpoint and the length of each successive interval so that for every $i = 0, 1, \ldots, 6$:

$$S_i = [\text{Left}_i; \text{Left}_i + \text{Length}_i]\,.$$

These sequences are defined in terms of the offsets and frequencies of the characters $c_0, c_1, \ldots, c_5$.

$$\begin{cases} \text{Left}_{i+1} = \text{Left}_i + \text{Left}_i \cdot \text{Offset}(c_i) \\ \text{Length}_{i+1} = \text{Length}_i \cdot \text{Frequency}(c_i) \end{cases}$$

For each encodable character the offsets and frequencies are defined in this table:

Character	Frequency	Offset
A	0.3	0.0
C	0.1	0.3
G	0.3	0.4
U	0.2	0.7
$	0.1	0.9

The only binary number β that belongs to $S_6 = [0.432976; 0.433030]$ is shown in answer **(D)**:

$$\beta = 0.011011101101100_2 = 0.4329834_{10}.$$

Question 2. Answer **(B)**
The only syntactically correct regular expression is

```
(\+371|\(\+371\)) [0-9]{8}
```

Question 3. Answer **(B)**
The probability that all 8 hash values will happen to be 1 is

$$0.7^8 = 0.05764801.$$

Question 4. (None of the answers is right)
We can add up the probabilities, using Poisson distribution formula:

$$P(X = k) = \frac{\lambda^k \cdot e^{-\lambda}}{k!}.$$

By adding up the first 5 values of this distribution we get this expression with $\lambda = 13.5$:

$$P(X < 5) = \sum_{k=0}^{4} \frac{\lambda^k \cdot e^{-\lambda}}{k!} \approx 0.000707.$$

Therefore the probability to get less than 5 raisins is 0.0707%.
Since all the answer variants were wrong, everyone gets full credit for this.

Question 5. Answer: 3
Three colors are sufficient as shown in the picture. But two colors are impossible (since the graph contains a triangle – a full graph K_3).

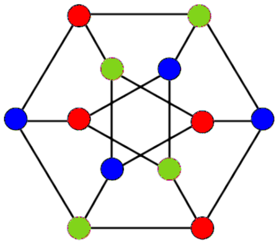


Figure 15. Dürer Graph colored in 3 colors.

This graph is planar (you can draw it without intersecting edges); it also has a famous polyhedron (Dürer solid); it was shown in an engraving made in 1514 called *Melencolia I*. See https://bit.ly/2KE2QZl.

Question 6. Answer: 0.125, 0.313, 0.250, 0.313
We write the vectors as they are multiplied with the 4×4 matrix. The components in the vector correspond to the pageranks (iterations 0, 1 and 2).

$$v_0 = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}; \quad v_1 = \begin{pmatrix} 1/8 \\ 1/2 \\ 1/8 \\ 1/4 \end{pmatrix}; \quad v_2 = \begin{pmatrix} 1/8 \\ 5/16 \\ 1/4 \\ 5/16 \end{pmatrix}.$$

Question 7. Answer **(D)**
The time complexity of such algorithm is $O(n^{\log_2 7})$ by Master's theorem. Since $\log_2 7 \approx 2.807355 < 2.808$, we have the time complexity in $O(n^{2.808})$ (it is the best estimate that is given among the arguments).

Question 8. Answer **(D)**
Since we can assume that one of the five strategies is optimal, it is sufficient to compare the strategies. We assume that Players A and B adopt one of the five suggested strategies plus one more simple strategy **(F)** (always guess number 2).
There are altogether 36 variants. For each pair of strategies we compute the payoff for the Player A. We want to find the strategy for Player A that beats any strategy chosen by B (or at least achieves a tie).

(A) $P(x = 1) = 1/3$, $P(x = 2) = 1/3$, $P(x = 5) = 1/3$.
(B) $P(x = 1) = 0$, $P(x = 2) = 1/2$, $P(x = 5) = 1/2$.
(C) $P(x = 1) = 1/2$, $P(x = 2) = 1/2$, $P(x = 5) = 0$.
(D) $P(x = 1) = 1/4$, $P(x = 2) = 1/2$, $P(x = 5) = 1/4$.
(E) $P(x = 1) = 2/6$, $P(x = 2) = 3/6$, $P(x = 5) = 1/6$.
(F) $P(x = 1) = 0$, $P(x = 2) = 1$, $P(x = 5) = 0$.

The strategies of Player A are shown in rows; strateges of Player B are shown in columns:

	(A)	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**
(A)	0	1/6	−1/6	0	−1/18	0
(B)	−1/6	0	0	0	0	1/2
(C)	1/6	0	0	0	0	−1/2
(D)	0	0	0	0	0	0
(E)	1/18	0	0	0	0	−1/6
(F)	0	−1/2	1/2	0	1/6	0

As you can see from the table, every strategy (except **(D)**) loses against some other strategy (there is at least one negative number on every line of the table). The only strategy that never loses is strategy **(D)**, i.e. picking the numbers $1, 2, 5$ with probabilities $\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$ respectively. (In fact, this is also Nash equilibrium – no other probabilistic strategy fares better than this.)
Note that without the strategy **(F)** we did not have enough evidence to see that strategies **(C)** and **(E)** are not optimal (because they only lose to strategy **(F)**). Moreover, the relationships between strategies are non-transitive:

- **(A)** beats **(B)**
- **(B)** beats **(F)**
- **(F)** beats **(C)** and **(E)**
- **(C)** and **(E)** beat **(A)**

The Nash equilibrium **(D)** does not beat anything in the list; it does not try to exploit "foolish" choices of the opponent.

Question 9. Answer **(C)**
This is known as Thue-Morse sequence. It is usually described at "marcro-level" - how to obtain new iterations of this sequence (by appending a new chunk of the previous iteration, where all letters have changed places). See https://bit.ly/3aExEnq.
But it is also possible to build Thue-Morse sequence at a "micro-level" (how to expand individual letters into pairs of letters):

$$\begin{cases} \texttt{A} \rightarrow \texttt{AB} \\ \texttt{B} \rightarrow \texttt{BA} \end{cases}$$

Note. https://bit.ly/2ypZ2rI shows other nice Lindenmayer images that can be created from the Thue-Morse sequence.

Question 10. Answer **(B)**
The number of hash values to be computed in order to have a $\frac{1}{2} = 50\%$ chance of a hash collision can be estimated using the square approximation. If we compute n values (and each value is a non-negative integer less than m), then

$$\frac{1}{2} \approx \frac{n^2}{m};\ \ n \approx \sqrt{m}.$$

If we have 100-bit hash values, then $m = 2^{100}$. And $n = \sqrt{2^{100}} = 2^{50}$.
We therefore need $n = 2^{50} \approx 1.1259 \cdot 10^{15}$. Now divide this number to convert from microseconds into years (divide by the number of milliseconds in a second; the number of seconds in an hour; the number of hours in a day; a number of days in year):

$$n = 1.1259 \cdot 10^{15} : 10^6 : 3600 : 24 : 365 \approx 35.70205.$$

Therefore the estimate is 35 years. (For SHA-256 the estimate would be 10.8 Septillions or about $10.8 \cdot 10^{24}$ years.) The collisions of secure hash functions exist (and by the Pigeonhole principle there should be many collisions for sufficiently short files). In fact, SHA-256 collisions are inevitable even when the size of the input file exceeds 256 bits (i.e. 32 bytes).

Question 11. Answer: **(C)**, **(E)**, **(G)**

- Hint 1 asks to consider the largest or the least frequently moved disk. (Going to the extremes, answer **(C)**).
- Hint 2 asks what should happen right before we can move the largest disk (Penultimate step, answer **(G)**).
- Hint 3 asks to see the symmetry in the problem (if you know how to move three disks from peg A to peg B, then you can also move them from peg A to peg C, and thus free the way for the fourth disk. (Looking for symmetries, answer **(E)**.

Other strategies are not directly used in the hints. Learning how to move three disks (before doing it with four disks) would mean making it easier (answer **(F)**), but it is already done by the learner.

Question 12. Answer:
`a787b563a07713fad9b68fb1d1370f5e`
It can be computed from the command-line:

```
md5sum hosts.txt
```

Question 13. Answer: 6

We can compute the best moves moving bottom up, starting with the leaves and computing the best payoff in every internal node (see Figure 16).

Question 14. Answer: 6, 2
There are $N = 6$ steps to go from P_0 to P_6. After that

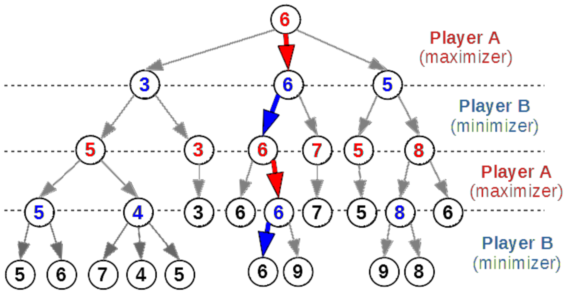


Figure 16. Minimax in a tree.

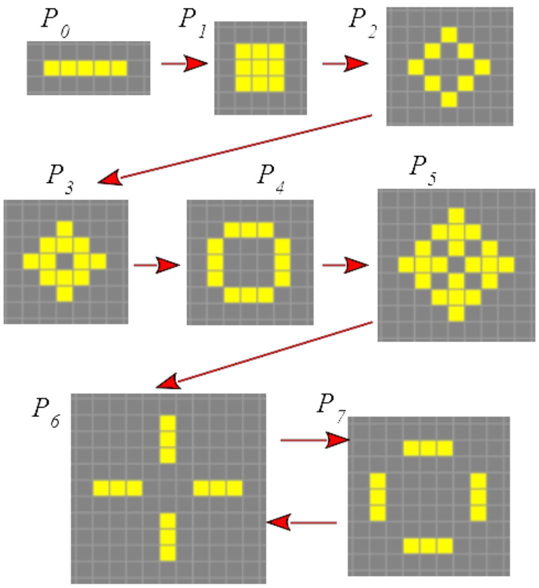


Figure 17. Conway Positions.

there is a periodic repeat of positions P_6 and P_7 with period $T = 2$ (see Figure 17).

Question 15. Answer: **(B)**

A theory which can prove a statement and its negation is called **not consistent**. (On the contrary, theories where this can never happen are called **consistent**).
BTW the first Gödel's Incompleteness Theorem states that any theory about integer numbers that is effectively axiomatizable and consistent should also be incomplete (i.e. there are true results that cannot be proven). So the theory $\mathcal{T}$ from our problem has a chance to be complete. But being inconsistent makes it completely useless (if a statement and its negation are both provable, then virtually anything can be proven, since both provable statements A and $\neg A$ also imply $(A \wedge \neq A) =$ `false`. But such `false` theorem would imply anything – whether it makes sense or not.

Question 16. Answer: 4

The full 19-digit credit card number is this:

`5573670544564505714`.

See https://bit.ly/2Y6NaWv for online Luhn check. The procedure is as follows:

- Drop the last digit from the number. (It is initially unknown in our case.)
- Reverse the digits.
- Multiply the digits in odd positions (1, 3, 5, etc.) by 2 and subtract 9 to all any result higher than 9.
- Add all the obtained numbers together
- The last (unknown) number is the amount that you would need to add to get a multiple of 10.

```
5,5,7,3,6,7,0,5,4,4,5,6,4,5,0,5,7,1
1,7,5,0,5,4,6,5,4,4,5,0,7,6,3,7,5,5
2,7,10,0,10,4,12,5,8,4,10,0,14,6,6,7,10,5
2,7,1,0,1,4,3,5,8,4,1,0,5,6,6,7,1,5
```

The sum of all digits on the last line is 66. By adding digit 4 this number becomes divisible by 10.

Question 17. Answer **(A)**

We should apply our intuition about natural language texts. If there are many identical sequences of five consecutive words, then they cannot appear by pure chance (there might be some common proverbs or short quotes, but they would not make 10% overlap.
Answer **(B)** is not credible – if random lookups in the hashtable lead to 5% matching, then 10% overlap cannot be explained by chance.
Answer **(C)** is not applicable either; rolling hash on words (especially, if it detects considerable number of matches) is more useful than rolling hash on characters – matching the characters is less robust and the hash windows are typically shorter.

Question 18. Answer 30
Truncated icosahedron (soccer ball graph) has the following number of faces, edges and vertices:

$$F = 32, E = 90, V = 60.$$

Since Hamilton graph visits all 60 vertices, it should also have 60 edges. For this reason, there are $E - V = 90 - 60 = 30$ edges that are not part of the Hamiltonian cycle.