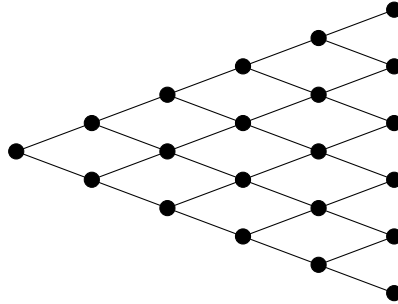


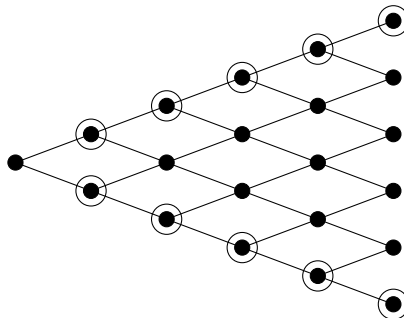
Computer Science Solutions Packet

1. What is the minimum number of times you have to take your pencil off the paper to draw the following figure (the dots are for decoration)? You're not allowed to draw over an edge twice.

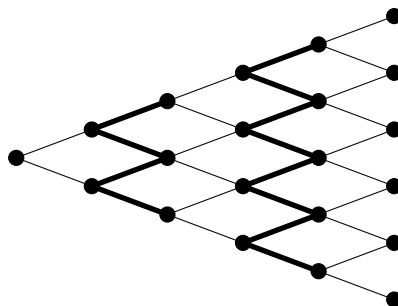


Proposed by Cody Johnson

Solution. The answer is $\boxed{5}$. Note that if a vertex has odd degree, then it must be the endpoint of some path (this is due to the what-goes-in-must-come-out rule of paths). There are 10 vertices of odd degree, so there must be at least 5 paths in total:



Furthermore, we can do it in 5 paths, illustrated below by the varying thicknesses:



2. We are given the following function f , which takes a list of integers and outputs another list of integers. (Note that here the list is zero-indexed.)

```

1: FUNCTION  $f(A)$ 
2:   FOR  $i = 1, \dots, \text{length}(A) - 1$ :
3:      $A[i] \leftarrow A[A[i]]$ 
4:    $A[0] \leftarrow A[0] - 1$ 
5:   RETURN  $A$ 

```

Suppose the list B is equal to $[0, 1, 2, 8, 2, 0, 1, 7, 0]$. In how many entries do B and $f(B)$ differ?

Proposed by David Altizio

Solution. It is possible to go through the entire algorithm step by step, but we note a few observations that make this easier.

- If $A[i] = i$, then the i^{th} element of A is unchanged. This is because $A[A[i]] = A[i]$, and so the algorithm feeds it $A[i]$, which by definition does not change.
- If $A[i] = j$ for some $j < i$ such that $A[j] = j$, then the i^{th} element of A is *also* unchanged. This is because $A[A[i]] = A[j] = j$, and so this element now contains j - which is what it contained before.

Thus the elements in positions 1, 2, 4, 6, and 7 do not change under this algorithm. On the other hand, note that the leading zero changes by line 4, the 8 in position 3 becomes a 0, and the two zeroes in positions 5 and 8 turn into negative numbers. As a result, exactly 4 elements between B and $f(B)$ are different.

3. In the following list of numbers (given in their binary representations), each number appears an even number of times, except for one number that appears exactly three times. Find the number that appears exactly three times. Leave the answer in its binary representation.

010111	000001	100000	011000	110101	100001
010100	011111	111001	010001	010100	101100
010001	011011	011111	011011	100000	000001
110011	001000	111101	100001	101100	110011
111111	011000	001000	101000	111111	101000
010111	100011	111001	100011	110101	011111
100000	010100	010001	101100	010111	011011
011000	111101	111111	100001	101000	100011
011011	010111	110011	111111	000001	010001
101000	111001	010100	110101	011000	110101
001000	000001	100000	111101	100011	001000
111001	110011	100001	011111	101100	

Proposed by Cody Johnson

Solution. We count that 1 appears as the first bit of these numbers an odd number of times, therefore it is unpaired so the first bit of the answer must be 1. Counting in a similar way for each successive bit, we have that the answer is 111101. The general algorithm is just to take the bitwise XOR of all of the numbers.

4. How many complete directed graphs with vertex set $V = \{1, 2, 3, 4, 5, 6\}$ contain no 3-cycles?

Proposed by Cody Johnson

Solution. Let $n = 6$. We show that there is a vertex with all edges pointing towards it (a sink). Suppose not, and let V be a vertex with the maximal number of edges pointing towards it. Then $V \rightarrow V'$ for some V' , and for each $U \rightarrow V$, we must have $U \rightarrow V'$ else $U \rightarrow V \rightarrow V' \rightarrow U$ is a 3-cycle. Therefore, V' has strictly more edges pointing towards it. There are n ways to label this sink, so we can delete it and multiply by the number of ways to do this of size $n - 1$. Since there is 1 such graph of size 1, there are $n!$ in total. Thus, the answer is 720.

5. Given a list A of n real numbers, the following algorithm, known as *insertion sort*, sorts the elements of A from least to greatest.

```

1: FUNCTION IS(A)
2:   FOR  $i = 0, \dots, n - 1$ :
3:      $j \leftarrow i$ 
4:     WHILE  $j > 0$  &  $A[j - 1] > A[j]$  :
5:       SWAP  $A[j], A[j - 1]$ 
6:        $j \leftarrow j - 1$ 
7: RETURN A

```

As A ranges over all permutations of $\{1, 2, \dots, n\}$, let $f(n)$ denote the expected number of comparisons (i.e., checking which of two elements is greater) that need to be made when sorting A with insertion sort. Evaluate $f(13) - f(12)$.

Proposed by Kyle Gettig

Solution. Note that when $i = k$ in the **FOR** loop, the first k elements of A are sorted and the $(k + 1)^{\text{th}}$ element is moved to the left until it reaches a number less than it, at which point it is inserted and the first $k + 1$ elements are sorted. Thus, the number of comparisons made at $i = k$ is equal to the number of elements with index less than k that are greater than $A[k]$, plus an additional count for the first such element less than $A[k]$ (if it exists) since that determines when the **WHILE** loop halts. If such an element doesn't exist, the extra comparison isn't added and the original $A[k]$ is at the beginning of the list. Now, let $g(A)$ denote the number of elements of A that are less than any element preceding it. From above, summing over all k we find $f(n) = \text{Inv}(A) + n - g(A)$, where $\text{Inv}(A)$ is the number of inversions in A .¹ Thus, $\mathbb{E}[f(n)] = \mathbb{E}[\text{Inv}(A)] + n - \mathbb{E}[g(A)]$.

It is not hard to see via linearity of expectation that $\mathbb{E}[\text{Inv}(A)] = \frac{1}{2} \binom{n}{2} = \frac{n(n-1)}{4}$, since for any two elements of A there is a $\frac{1}{2}$ probability that they are in increasing order or decreasing order. Now it suffices to determine $\mathbb{E}[g(A)]$, and furthermore we need only compute the probability that the k^{th} smallest element of A is smaller than any element preceding it. For the largest element, this is possible only if it is the first element in A . This occurs with probability $\frac{1}{n}$. For the second largest element, it must either be the first element, or the second element with the largest element before it. This occurs with probability $\frac{1}{n} + \frac{1}{n(n-1)} = \frac{1}{n-1}$. Induction shows that the probability the k^{th} smallest element of A is $\frac{1}{k}$, and then by linearity of expectation we conclude $\mathbb{E}[g(A)] = \sum_{k=1}^n \frac{1}{k} = H_n$, where H_n is the n^{th} harmonic number.

Now, $f(n) = \frac{n(n-1)}{4} + n - H_n$, and so $f(n) - f(n-1) = \frac{n+1}{2} - \frac{1}{n}$. When $n = 13$, the desired value is $\boxed{\frac{97}{14}}$.

6. Define a self-balanced tree to be a tree such that for any node, the size of the left subtree is within 1 of the size of the right subtree. How many balanced trees are there of size 2046?

Proposed by Cody Johnson

Solution. Let $S(n)$ denote the number of self-balanced trees of size n . Note that $S(2n + 1) = S(n)^2$ since a tree is self-balanced if and only if both subtrees of the root are self-balanced trees of size n . Also, $S(2n) = 2S(n)S(n-1)$ since a tree is self-balanced if and only if both subtrees of the root are self-balanced and either the left subtree is of size $n-1$, and the right subtree is of size n , or the left subtree is of size n , and the right subtree is of size $n-1$. From these recurrences, it's easy to show by induction that $S(2^n) = 2^n$ and $S(2^n - 1) = 1$, and hence $S(2^n - 2) = 2^{n-1}$. Thus, the answer is $S(2046) = S(2^{11} - 2) = 2^{10} = \boxed{1024}$.

7. You are presented with a mystery function $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ which is known to satisfy

$$f(x + 1, y) > f(x, y) \quad \text{and} \quad f(x, y + 1) > f(x, y)$$

for all $(x, y) \in \mathbb{N}^2$. I will tell you the value of $f(x, y)$ for \$1. What's the minimum cost, in dollars, that it takes to compute the 19th smallest element of $\{f(x, y) \mid (x, y) \in \mathbb{N}^2\}$? Here, $\mathbb{N} = \{1, 2, 3, \dots\}$ denotes the set of positive integers.

Proposed by Cody Johnson

¹That is, the number of pairs $i < j$ with $A[i] > A[j]$.

Solution. Let $n = 19$. First, note that if $(a - 1) + (b - 1) > n$,

$$f(a, b) > f(a - 1, b) > \cdots > f(1, b) > f(1, b - 1) > \cdots > f(1, 1)$$

so $f(a, b)$ cannot be the n th smallest element. Furthermore, $f(1, 1)$ cannot be the n th smallest element.

Now suppose $a + b \leq n + 2$ and $(a, b) \neq (1, 1)$. Without loss of generality, assume $a \neq 1$. Then consider the function

$$f(x, y) = \begin{cases} 1 + y & x = 1 \wedge y \leq n - 1 \\ (x - a) + (y - b) + n + 1 & x \geq a \wedge y \geq b \\ x + y & x + y \leq n \\ x + y + 1 & x + y \geq n + 1 \end{cases}$$

It's easy to verify that f satisfies the condition that

$$f(x + 1, y), f(x, y + 1) > f(x, y)$$

for all $(x, y) \in \mathbb{N}^2$. Furthermore, the $n - 1$ values $2, \dots, n$ appear as $f(1, y)$ for $1 \leq y \leq n - 1$, so $f(a, b) = n + 1$ is indeed the n th smallest element of $\{f(x, y) \mid (x, y) \in \mathbb{N}^2\}$. Finally, it's unique, so no other $(x, y) \in \mathbb{N}^2$ satisfies $f(x, y) = n + 1$. Therefore, all algorithms must check the value of $f(a, b)$. Thus, all algorithms must check at least $\frac{(n+1)(n+2)}{2} - 1 = \frac{n^2+3n}{2}$ values. Therefore, the answer is $\frac{19(22)}{2} = \boxed{209}$.

8. We have a collection of 1720 balls, half of which are black and half of which are white, aligned in a straight line. Our task is to make the balls alternating in color along the line. The following greedy algorithm accomplishes that task:

- 1: **FOR** i **IN** $[2, 3, \dots, 2n]$
- 2: **IF** balls $i - 1$ and i have the same color:
- 3: $j \leftarrow$ smallest index greater than i for which balls $i - 1$ and j have different colors
- 4: swap balls i and j

Given a configuration C , let $\hat{\sigma}(C)$ denote the number of swaps the greedy algorithm takes, and let $\sigma(C)$ denote the minimum number of swaps actually necessary to perform the task. Find the maximum value over all configurations C of $\hat{\sigma}(C) - \sigma(C)$.

Proposed by Cody Johnson and Victor Xu

Solution. Let there be $2n$ balls. We first claim $\hat{\sigma}(C) - \sigma(C) \leq n - 2$. The greedy algorithm does at most $n - 1$ swaps since, after each swap, if the iteration is at index i then the color of the balls at positions i and $i + 1$ must be different, and there are only $2n - 2$ potential spots where we could ever swap something (note that we never swap at the end). Trivially, $\sigma(C) \geq 1$ if C is not already alternating, and so $\hat{\sigma}(C) - \sigma(C) \leq n - 1 - 1 = n - 2$, as desired.

Now we show $n - 1$ swaps is achievable. Consider the configuration $BBWBWBWB \dots BWBWW$, where the first two balls are the same color, then the colors alternate, and the last two balls are the same color. It's easy to verify that for this configuration we have $\hat{\sigma}(C) - \sigma(C) = n - 2$. Thus $n - 2$ is the maximum achievable value.

Since $2n = 1720$, it follows that $n - 2 = \boxed{858}$.

9. Alice thinks of an integer $1 \leq n \leq 2048$. Bob asks k true or false questions about Alice's integer; Alice then answers each of the questions, but she may lie on at most one question. What is the minimum value of k for which Bob can guarantee he knows Alice's integer after she answers?

Proposed by Patrick Lin

Solution. Equivalently, we consider *error-correcting codes*; note that integers between 1 and 2048 are 11-bit integers, so we wish to find a minimal k such that we can find a map $\varphi : \{0, 1\}^{11} \rightarrow \{0, 1\}^k$ where φ can correct for at most one error (i.e. the possibility that Alice lies on a question). For each integer $i \in \{0, 1\}^{11}$, consider $\varphi(i)$. If we wish for φ to be able to correct for one error, then all k strings that are of Hamming

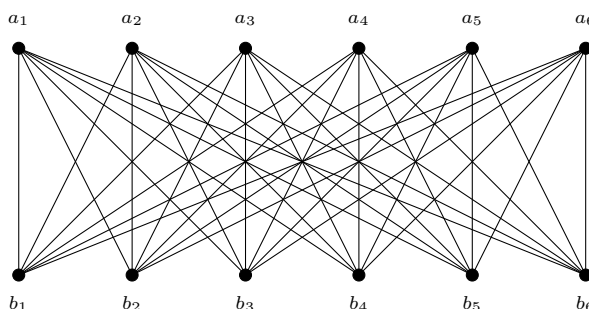
distance 1 from $\varphi(i)$ (i.e. they differ in one place) must be associated with i and no other integer. Hence, each of the 2048 integers need to be associated with disjoint sets of size $k + 1$, for if they were not disjoint, then we would not be able to uniquely recover the original integer. This gives the inequality

$$2048 \cdot (k + 1) \leq 2^k,$$

since there are only 2^k integers in $\{0, 1\}^k$ and we need at least $2048 \cdot (k + 1)$ integers for there even to be a chance of such a map φ existing. The equality case is at $k = \boxed{15}$.

The explicit construction of such a map φ is known as the *Hamming (15,11) code*. It is difficult to explain succinctly but can be found at https://en.wikipedia.org/wiki/Hamming_code.

10. How many distinct spanning trees does the graph below have? Recall that a *spanning tree* of a graph G is a subgraph of G that is a tree and containing all the vertices of G .



Proposed by Andrew Kwon

Solution. Note that the graph given is the complete bipartite graph on 12 vertices partitioned evenly; we claim the answer is 6^{10} , and that in general for $K_{n,n}$ the answer is n^{2n-2} . Suppose the upper half of the vertices are labelled a_1, \dots, a_n and the lower half of the vertices are labelled b_1, \dots, b_n . We provide an explicit bijection between spanning trees on $K_{n,n}$ and sequences of the form $a_{i_1}, b_{j_1}, a_{i_2}, b_{j_2}, \dots, a_{i_{n-1}}, b_{j_{n-1}}$, where none of the terms are necessarily distinct. Evidently there are n^{2n-2} such sequences.

We refer to vertices of a spanning tree with degree 1 as leaves, and we claim that there must exist a leaf among the a_i as well as among the b_i . Evidently a tree on $2n$ vertices has $2n - 1$ edges, and therefore the sum of all of the degrees of the vertices in the graph is $4n - 2$. Also, the sum of the degrees of the a_i and the sum of the degrees of the b_i are equal since every edge in the spanning tree connects some a_i and some b_j . Thus, the sums of the degrees for each is $2n - 1$, and it is impossible that $\deg a_i, \deg b_i \geq 2$ for all $1 \leq i \leq n$.

Now we describe an algorithm to encode spanning trees as sequences of the claimed form, as well as the inverse algorithm to decode them.

To encode a tree as a sequence,

- (a) Consider the b_{i_1} with minimal index that is a leaf; it is connected to some a_{i_1} , which we shall take to be the first term of the sequence and delete b_{i_1} along with its edge.
- (b) Then, consider the a_{j_1} with minimal index that is a leaf; it is connected to some b_{j_1} , which we shall take to be the next term of the sequence and delete a_{j_1} along with its edge.

Note that our tree now has $2n - 2$ vertices, and the above argument demonstrates we again have one leaf among the upper and lower vertices. Thus we may repeat steps 1 and 2 on our smaller graph until there is one edge left, then stop.

As the above procedure ends with 2 vertices, it deletes $2n - 2$ vertices and generates a sequence of length $2n - 2$ of the desired form. Furthermore, leaves of the tree never appear in the sequence. With this we present the inverse algorithm to construct a tree from a sequence.

Let $A = \{a_{i_1}, \dots, a_{i_{n-1}}\}, B = \{b_{j_1}, \dots, b_{j_{n-1}}\}$. To decode a tree from a sequence $a_{i_1}, b_{j_1}, \dots, a_{i_{n-1}}, b_{j_{n-1}}$, for $k = 1, \dots, n - 1$,

COMO 2017

- (a) Consider the b_{i_k} with minimal index that is not in B . Add (a_{i_k}, b_{i_k}) to the tree and b_{i_k} to B .
- (b) Consider the a_{j_k} with minimal index that is not in A . Add (a_{j_k}, b_{j_k}) to the tree and a_{j_k} to A .

After iterating through $k = 1, \dots, n - 1$, add $(a_{i_{n-1}}, b_{j_{n-1}})$ to the tree and stop.

As each step of the encoding algorithm is reversed by the corresponding step in the decoding algorithm, e.g., step (a) of the encoding algorithm is undone by step (a) of the decoding algorithm, it is clear that these procedures lead to a bijection between the spanning trees of $K_{n,n}$ and strings of length $2n - 2$ from an alphabet on n characters (where it suffices to consider only the indices of the a_i, b_i). Thus, in the case of $K_{6,6}$ we find there are $\boxed{6^{10}}$ such spanning trees.