


<https://swayam.gov.in>

https://swayam.gov.in/nc_details/NPTEL

jigarpandya.ce@gmail.com ▾

NPTEL (<https://swayam.gov.in/explorer?ncCode=NPTEL>) » Getting Started With Competitive Programming (course)



Click to register
for Certification
exam

(https://examform.nptel.ac.in/2023_10/exam_form/dashboard)

If already
registered, click
to check your
payment status

Course
outline

How does an
NPTEL
online
course
work? ()

Week 0 ()

☐ Practice:
Week 0 :
Assignment
(Non Graded)
(assessment?
name=238)

Week 0 : Assignment (Non Graded)

Assignment not submitted

Note : This assignment is only for practice purpose and it will not be counted towards the Final score

1) How many ones does the following procedure print when run with input n ?

1 point

```
Ones(n) :
    if n = 0:
        print 1
    else:
        for i = 1 to 2^n:
            Ones(n-1)
```

Read the upper bound on the for loop as 2^n . Consider the options below:

A. $\Theta(2^{n^2})$

B. $2^{\Theta(n^2)}$

C. $\Theta(2^{n(n+1)/2})$

D. $\Theta(n \cdot 2^n)$

E. $\Theta(n^2 2^n)$

- ☐ Option A
☐ Either B or C (both are valid)
☐ Option D
☐ Option E

2) How many connected components result after performing the following sequence of union operations on a set of 10 items?

1-2, 3-4, 5-6, 7-8, 7-9, 2-8, 0-5, 1-9

1 point

3) How many bit strings (i.e, sequences over $\{0,1\}$) of length 18 have no consecutive ones? (You may write a program to answer this question, although a calculator should also suffice.)

1 point

4) Mark the correct options.

1 point

☐

The functions $\{2^{\log n}, n \log n, 2^n, n!\}$ are listed in the increasing order of their asymptotic growth rate.

☐

The functions $\{n \log n, 2^{\log n}, 2^n, n!\}$ are listed in the increasing order of their asymptotic growth rate.

☐

The functions $\{n \log n, 2^{\log n}, n!, 2^n\}$ are listed in the increasing order of their asymptotic growth rate.

☐

$f(n) = O(f(n^2))$.

☐

For any constant k , $n^k \in O(n^{k+1})$ but $n^{k+1} \notin O(n^k)$.

☐

For any constant k , $n^{k+1} \in O(n^k)$ but $n^k \notin O(n^{k+1})$

5) We have an exponential time algorithm for problem A, and problem A reduces in polynomial time to problem B. From this we can conclude that:

1 point

- ☐ B has an exponential time algorithm.
☐ B cannot have a polynomial time algorithm.
☐ A cannot have a polynomial time algorithm.

☐ None of the above.

6) Which of the following is a stable sorting algorithm that takes the least time when applied to an almost sorted array ? (Recall that a sorting algorithm is stable if two objects with equal keys appear in the same order in sorted output as they appear in the input array to be sorted). **1 point**

- ☐ Quick Sort
☐ Insertion Sort
☐ Selection Sort
☐ Merge Sort

7) Which of the following statements is true?

1 point

Statement 1: For every graph G and every maximum flow on G , there always exists an edge such that increasing the capacity on that edge will increase the maximum flow that's possible in the graph.

Statement 2: Suppose the maximum (s, t) -flow of some graph has value f . Now we increase the capacity of every edge by 1. Then the maximum (s, t) -flow in this modified graph will have value at most $f + 1$.

- ☐ Statement 1
☐ Statement 2
☐ Both statements
☐ Neither statement

8) Suppose you have a set of N people who all do not know each other, and you have to process a sequence of queries of the following kinds: **1 point**

Type 1: MakeFriends (x, y) — x and y become friends with each other

Type 2: AreFriends (x, y) - output 1 if x and y are friends, and output 0 otherwise

In particular, for every pair of people, we need to maintain a state indicating whether they are friends or not, and update this state based on queries of type 1, and report the state for queries of type 2.

Since you have learned about the disjoint sets data structures in your algorithms course, you decide to store the set of N people as N singleton sets to begin with. For each query of type 1, you perform a union operation, and for each query of type 2, you perform a find operation.

What can you say about this algorithm?

- ☐ It may not always produce an accurate answer.
☐ It works with $O(\lg^* n)$, $O(\lg^* n)$ amortized response time to both queries.
☐ It works with $O(n)$, $O(n)$ worst-case response time to both queries.
☐ The correctness of the approach depends on how DSU is implemented.

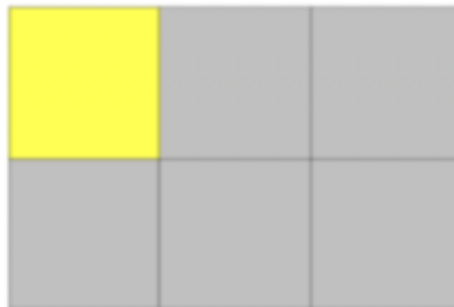
Lights Out is a one-person game played on a rectangular lattice of lamps which can be turned on and off. A move consists of flipping a "switch" inside one of the squares, thereby toggling the on/off state of this and all four vertically and horizontally adjacent squares. Starting from a randomly chosen light pattern, the aim is to turn all the lamps off.

9) If the rectangular lattice you are given is a 1×5 grid, as shown below, what is the **0.5 points** smallest number of moves you need to turn off all the squares?



- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ Impossible to turn off all squares

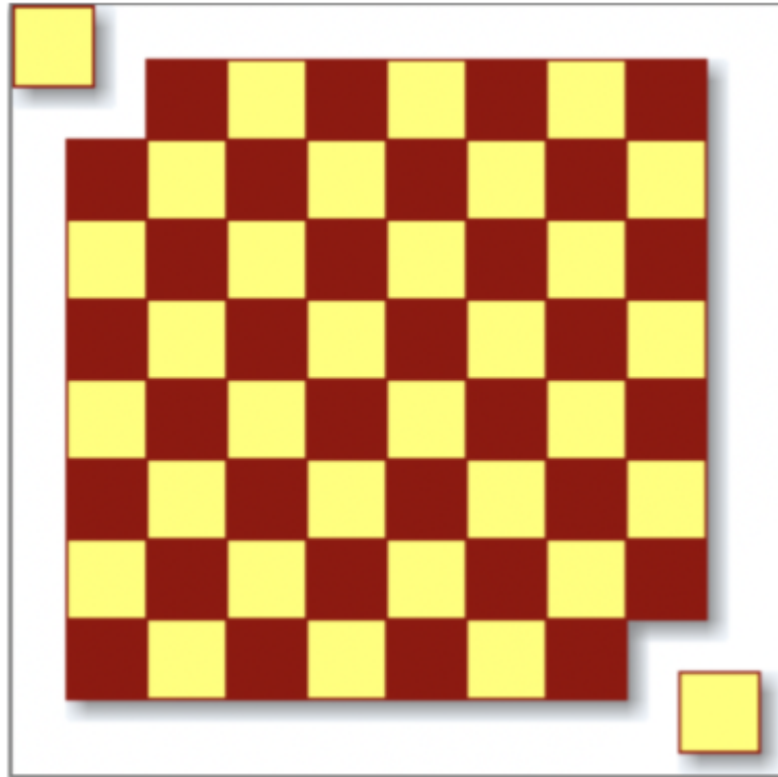
10) If the rectangular lattice you are given is a 2×3 grid, as shown below, with the **0.5 points** top-left corner on to begin with, in how many moves can you turn off all the squares?



- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ Impossible to turn off all squares

11) We have a chessboard and 32 dominoes. Each domino is of such size that it **1 point** exactly covers two adjacent squares on the board. The 32 dominoes therefore can cover all 64 of the chessboard squares. But now suppose we cut off two squares at diagonally opposite corners of the board and discard one of the dominoes.

Is it possible to place the 31 dominoes on the board so that all the remaining 62 squares are covered?



☐ Yes

☐ No

12) Suppose we have a simple, undirected, connected graph G where all edges have non-negative weights, and in fact, the weight of each edge is either 1 or 20.

Consider the following subgraph H of G — the vertex set of H is the same as the vertex set of G and the edge set of H consists of all the edges from G whose weight is one.

Suppose H has three connected components with 10, 15, and 30 vertices. Then the weight of any minimum spanning tree of G is:

1 point

13) When an array is to be sorted, it may happen that some data values start out being **1 point** in the same position where they should end up. For example, in the array which is originally:

45, -4, 32, 0

the 32 is right where it will be in the final sorted output: .

-4, 0, 32, 45

But as a particular sorting algorithm operates, it might (depending on the algorithm) move such an element out of the position where it belongs and move it back eventually. Let's say that a sorting algorithm *respects fixed points* if it never moves an element that is in its proper position, on any input.

Which of the following statements are true? In the options below, the implementation of selection sort is based on swapping the minimum element into its proper place in the current iteration, and does not involve the use of heaps. You may also assume that the input array has no duplicate elements.

- ☐ Insertion sort does not respect fixed points but selection sort does.
- ☐ Selection sort does not respect fixed points but insertion sort does.
- ☐ Neither insertion sort nor selection sort respects fixed points.
- ☐ Both insertion sort and selection sort respect fixed points.

14) You are at a party where any two people have an odd number of mutual friends at the party. You see an even number of people other than you. What is the parity of the total number of attendees? Note that we follow the convention that nobody is friends with themselves. **1 point**

- ☐ Even
- ☐ Odd
- ☐ Can be either odd or even
- ☐ Such a party cannot exist

15) In a party consisting of 5 couples, people who knew each other shook hands (pre-covid era) while those who did not know each other just greeted each other. One of them, X , observed: "If you don't count me, then there are no two people present who shook hands the same number of times". How many times did the wife of X shake hands? **1 point**

(Nobody shook hands with himself or his spouse!)

- ☐ 3
- ☐ 4
- ☐ 0
- ☐ Can't be determined.

16) Given an undirected graph G , we would like to color its vertices red and blue in such a way that the total number of edges between the red vertices is as close as possible to the total number of edges between the blue vertices. In other words, suppose the vertices of graph G are partitioned into the sets A and B , and the number of edges with both endpoints in A is denoted as a , and the number of edges with both endpoints in B is denoted as b . Then, let us define the "quality" of the partition (A, B) as $|a - b|$ and denote this by $q(A, B)$. The goal is then to find a partition (A, B) that minimizes $q(A, B)$. This is equivalent to:

- ☐ minimizing the number of edges with exactly one endpoint in A and one endpoint in B
- ☐ maximizing the number of edges with exactly one endpoint in A and one endpoint in B
- ☐ minimizing the difference between the sums of the degrees of the vertices in A and B
- ☐ None of the above

Check Answers and Submit

