Course outline

About NPTEL ()

Week 1 : Assignment 1

Assignment not submitted

Due date: 2025-02-05, 23:59 IST.

1) Consider the following recursive function where `D` is an array and `s1` and `s2` are indices in the range `0` to `length(D) - 1` : ***1 point***

```
function fun(D, s1, s2):
    if s1 < s2 then
        swap(D[s1],D[s2])
        fun(D, s1 + 1, s2 - 1)
```

What does this function `fun()` do?

How does an NPTEL online course work? ()

Week 0 ()

Week 1 ()

- Welcome and Initial Setup (unit?unit=17&lesson=18)
- Reversort (unit?unit=17&lesson=19)
- Engineering Reversort (unit?unit=17&lesson=20)
- Number Game (unit?unit=17&lesson=21)
- Will It Stop? (unit?unit=17&lesson=22)
- Week 1 Feedback Form: Getting Started with Competitive Programming (unit?unit=17&lesson=24)
- Practice: Week 1: Assignment 1(Non Graded) (assessment?name=487)
- Week 1 Practice Programming Assignment 1 (/noc25_cs36/progassignment?name=489)

- It finds the smallest element in D from index s1 to s2, both inclusive.
- It reverses the list D between indices s1 and s2, both inclusive
- It performs a merge sort in-place on this list D between indices s1 and s2, both inclusive
- It swaps the elements in D at indices s1 and s2, and leaves the remaining elements unchanged.

2) Consider the following recursive function where `L` is an array and `i` is an index in the range `0` to `length(L)` : ***1 point***

```
function compute(L, i):
    if i == length(L) then
        return 0
    else if L[i] is odd then
        return 1 + compute(L, i + 1)
    else
        return compute(L, i + 1)
```

What does this function `compute()` do?

- It returns the number of even elements in the list `L` from index `i` to the end of the list.
- It returns the sum of even elements in the list `L` from index `i` to the end of the list.
- It returns the number of odd elements in the list `L` from index `i` to the end of the list.
- It returns the sum of even elements in the list `L` from index `i` to the end of the list.

- Week 1 Practice Programming Assignment 2 (/noc25_cs36/progassignment?name=490)
- Week-1 Programming Assignment Q1 (/noc25_cs36/progassignment?name=483)
- Week 1 Programming Assignment Q2 (/noc25_cs36/progassignment?name=484)
- **Quiz: Week 1 : Assignment 1 (assessment?name=485)**

Week 2 ()

Week 3 ()

Download Videos ()

Transcripts ()

Books ()

3) Consider the following code: ***1 point***

```
function foo(n):
    i = 1
    result = 0
    while (i * i <= n) do
        result = result + i
        i = i + 1
    end while
    return result
```

What is the running time complexity of the given function?

- ○ $O(1)$
- ○ $O(logn)$
- ○ $O(n)$
- ○ $O(\sqrt{n})$

4) Consider the following `Reversort` algorithm discussed in lecture ***1 point***

```
1  Reversort(L):
2    for i := 1 to length(L) - 1:
3      j := position with the minimum value in L between i and length(L),
   inclusive
4      Reverse(L[i..j])
```

For which of the following input arrays of size N=8, the cost C will be the **minimum** when applying the `reversort` algorithm?

- ○ [1, 7, 6, 5, 4, 3, 2, 8]
- ○ [1, 2, 6, 7, 5, 3, 4, 8]
- ○ [1, 7, 6, 3, 4, 5, 2, 8]
- ○ [2, 1, 4, 3, 6, 5, 8, 7]

5) Consider the problem `Reversort` discussed in the lecture. What cost will be added in the fourth iteration if the input array is `[3, 5, 7, 2, 6, 4, 8, 1]` ? ***1 point***

- ○ 2
- ○ 5
- ○ 4
- ○ 1

6) Consider the problem **Reversort** discussed in the lecture. What would be the minimum of the input array possible for **Cost** 90?

1 point

7) Consider the problem **Number Game** discussed in the lecture. Suppose now it's Arya's turn. Which of the following is/are the winning position for Arya? ***1 point***

- ☐ A = B
- ☐ A >= (34/21)B

- [ ] A >= (89/55)B
- [ ] A <= (144/89)B

8) The **game of "40"** is played with two players who take turns saying a number, and neither player is allowed to pass. The first player says "1," and each player in turn increases the number by 1, 2, or 3, but the number may not exceed 40. The player forced to say "40" loses. If the first player starts with "1," which of the following will be the **winning strategy** for the second player? ***1 point***

- ○ The second player should always say numbers that are exact multiples of 4 (e.g., 4, 8, 12, 16, etc.)
- ○ The second player should always ensure that the current number is one less than a multiple of 4 (e.g., 3, 7, 11, 15, etc.)
- ○ The second player should always add exactly 3 to the previous number
- ○ The second player should always add exactly 2 to the previous number

9) onsider the code given in the lecture "Will It Stop?". If we replace the term $3n+3$with $2n+1$, then the code will look like below: ***1 point***

```
while (n > 1)
{
    if (n % 2 == 0)
    {
        n = n/2;
    }
    else
    {
        n = 2 * n + 1;
    }
}
```

Which of the following options is correct? Consider that $1 < n \leq 10^3$?

- [ ] The code will stop for all odd `n` .

- [ ] The code will stop for all even `n` .
- [ ] The code will stop for all values of `n` in the given range
- [ ] The code will never stop for any odd `n` .
- [ ] The code will stop for all that are powers of 2

10) Consider a problem where you begin with a stack of `n` boxes, and you make a sequence of moves. In each move, you divide one stack of boxes into two non-empty stacks. The game ends when you have `n` stacks, each containing a single box. You earn points for each move; in particular, if you divide one stack with boxes `a + b` into two stacks with boxes `a` and `b` , then you get score `a * b` points for that move. Your overall score is the sum of the points that you earn for each move, and it should be maximum in all possible strategy to dividing the stack.

For example `n=8`

```
One possible way of dividing:-
8
6 2 (Points for that move is 6 * 2 = 12)
3 3 2 (Points for that move is 3 * 3 = 9)
2 1 3 2 (Points for that move is 2 * 1 = 2)
2 1 2 1 2 (Points for that move is 2 * 1 = 2)
1 1 1 2 1 2 (Points for that move is 1 * 1 = 1)
1 1 1 1 1 1 2 (Points for that move is 1 * 1 = 1)
1 1 1 1 1 1 1 1 (Points for that move is 1 * 1 = 1)
Total Score = 12 + 9 + 2 + 2 + 1 + 1 + 1 = 28
```

What would be the maximum score if `n=20` ?

1 point

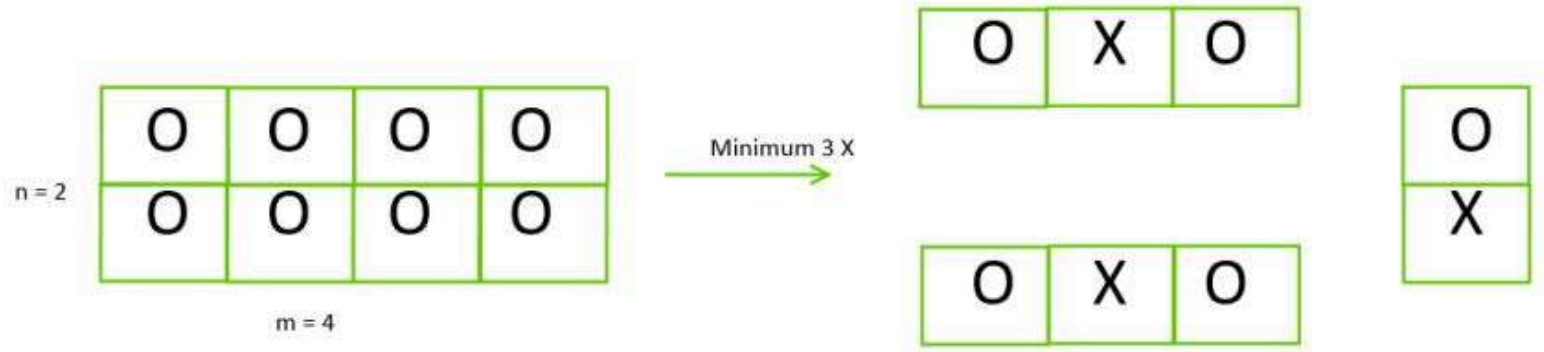
11) Given a wooden piece, a grid containing **n** rows and **m** columns, each 1 x 1 square containing **O** written inside it. You can cut the
or any other rectangular piece obtained during the cutting into two new pieces along the grid lines. You will obtain a certain number of rec

pieces after doing the cutting. **1 x 1 is a square, you cannot treat it as a rectangle**.

Your task is to design each rectangular piece obtained in such a way that any pair of adjacent cells have different symbols. What would be the minimum number of cells you need to put an **X** on in an **n x m** grid to achieve the desired result?

Symbols : **O** and **X**

Example:- For n = 2 and m = 4, minimum number of change **O** to **X** is 3



What would be the minimum number of cells you need to put an **X** on in a **4 x 7** grid to achieve the desired result?

1 point

You may submit any number of times before the due date. The final submission will be considered for grading.

Submit Answers