



Ain Shams University
Faculty of Engineering
CSE332s: Design and Analysis of Algorithm – Spring 2024
PROJECT REQUIREMENTS

This project is a group project with each group has 10 or 12 students. Each team must do the following Tasks:

Task 1

Devise an algorithm for the following task: given a $2^n \times 2^n$ ($n > 1$) board with one missing square, tile it with right trominoes of only three colors so that no pair of trominoes that share an edge have the same color. Recall that the right tromino is an L-shaped tile formed by three adjacent squares.

Use dynamic programming to solve this problem.

Task 2

Is it possible for a chess knight to visit all the cells of an 8×8 chessboard exactly once, ending at a cell one knight's move away from the starting cell? (Such a tour is called closed or re-entrant. Note that a cell is considered visited only when the knight lands on it, not just passes over it on its move.)

If it is possible design a greedy algorithm to find the minimum number of moves the chess knight needs.

Task 3

There is a row of n security switches protecting a military installation entrance. The switches can be manipulated as follows:

- (i) The rightmost switch may be turned on or off at will.
- (ii) Any other switch may be turned on or off only if the switch to its immediate right is on and all the other switches to its right, if any, are off.
- (iii) Only one switch may be toggled at a time.

Design a Dynamic Programming algorithm to turn off all the switches, which are initially all on, in the minimum number of moves. (Toggling one switch is considered one move.) Also find the minimum number of moves.

Task 4

There are eight disks of different sizes and four pegs. Initially, all the disks are on the first peg in order of size, the largest on the bottom and the smallest on the top.

Use divide and conquer method to transfer all the disks to another peg by a sequence of moves. Only one disk can be moved at a time, and it is forbidden to place a larger disk on top of a smaller one.

Does the Dynamic Programming algorithm can solve the puzzle in 33 moves? If not then design an algorithm that solves the puzzle in 33 moves.

Task 5

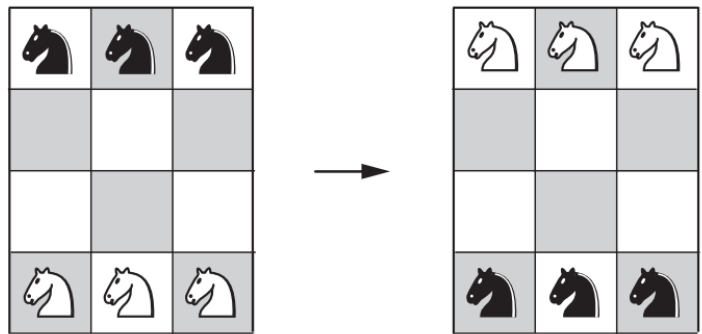
There are n coins placed in a row. The goal is to form $n/2$ pairs of them by a sequence of moves. On the first move a single coin has to jump over one coin adjacent to it, on the second move a single coin has to jump over two adjacent coins, on the third move a single coin has to jump over three adjacent coins, and so on, until after $n/2$ moves $n/2$ coin pairs are formed. (On each move, a coin can jump right or left but it has to land on a single coin. Jumping over a coin pair counts as jumping over two coins. Any empty space between adjacent coins is ignored.) Determine all the values of n for which the problem has a solution and design an algorithm that solves it in the minimum number of moves for those n 's.

Design greedy algorithm to find the minimum number of moves.

Task 6

There are six knights on a 3 × 4 chessboard: the three white knights are at the bottom row, and the three black knights are at the top row.

Design an iterative improvement algorithm to exchange the knights to get the position shown on the right of the figure in the minimum number of knights moves, not allowing more than one knight on a square at any time.



Task 7

A computer game has a shooter and a moving target. The shooter can hit any of $n > 1$ hiding spot located along a straight line in which the target can hide. The shooter can never see the target; all he knows is that the target moves to an adjacent hiding spot between every two consecutive shots. Design a Dynamic Programming algorithm that guarantees hitting the target.

Task 8

If you have 50 boxes that contains 50 pieces of metal all of the same known weight. one of these boxes contains fake metal pieces that weigh 1 kilogram less than the pieces in the rest of the boxes. You can use a digital scale only once to find this fake box. Design a brute force algorithm to solve this problem.

The report **MUST** contain each of the following items (items 2 to 8 should be provided for each task separately):

1. Cover page that shows the group names, college name, program name, course code, course name ... etc.
2. Detailed assumptions.
3. Problem description.
4. Detailed solution including the pseudo-code and the description of the steps of your solution.
5. Complexity analysis for the algorithm.
6. A comparison between your algorithm and at least one other technique that can be used to solve the problem.
7. Sample output of the solution for the different cases of the technique with proper description for the output.
8. Conclusion.
9. References that should be clearly cited inside the document.
10. Any additional needed sections.

Project Deliverable

All deliverable must be submitted on the LMS, no deliverable will be accepted by any other means. The following are required to be delivered by the due date:

1. Project code using any programming language. It must be submitted as a zip/rar archive.
2. A presentation (in .pptx format) of the different phases of the project. Each group will do the presentation and project demo via Microsoft-Team after submitting the project.
3. Project document (in .docx format) that contains at least the above-mentioned sections.

The following instructions **MUST** be taken into consideration while doing your project

- Use consistent document format (font sizes, titles, subtitles, captions, paragraph formatting ... etc.). Recommended font sizes are: main title 14pt, subtitles 12pt, main text 12pt, and captions 10 pt. Recommended font type is bold "Calibri" for titles and subtitles, and regular "Calibri" for all other texts. Recommended spaces before and after paragraphs are 12pt before and 6pt after each paragraph, and 1.5 spacing is highly recommended. Justified paragraphs from both sides are also recommended.
- Figures and tables must be centred in the pages, and they should be numbered separately. Each figure must have a caption that appears below the figure, and each table must have a title above it.
- Pages must be numbered consistently except for the cover page.

- Table of Contents must be included in your document that shows the titles and sub-titles of your report with the corresponding page numbers.
- List of figures and list of tables must be included in your document if you have figures/tables in your document.
- All reports must be written in English, always avoid typos and grammatical errors.
- All submitted files will undergo plagiarism check.
- All project deliverables must be uploaded to the LMS, no hardcopy is accepted, and please do not send your project deliverables by email. No other means of submission will be accepted.