Project ideas 2022

The project is to be completed by a small team of **2-3** people. Your task is to come up with one project proposal all by yourself. This is an individual task, you can share your idea in **Slack** to attract your prospective team members. You can take ideas from previous project topics, your imagination + omnipotent Google. You need to present your project as a poster presentation (physically) on the tentative date of **Friday, January 20th, 10 am-1 pm**

Project teams - Algorithmics 2022

Please, fill in your project - title, team members, and link to your own project environment (code repo, Drive Doc, or whatever)

TEMPLATE (Please, don't edit):

Title
Team: Member 1; Member 2
2-3 sentences describing the project idea and goals
Link: where is project relevant information during the project

---

Add your project topic and team here:

EXAMPLE PROJECT:

Building artificial general intelligence using deep K-nearest neighbor algorithm
Team: Jaak Vilo, Kallol Roy, Joonas Puura

First we want to develop a new algorithm by making the KNN algorithm deeper. Make our own version of AlphaGo by training a K-nearest neighbor algorithm. Use transfer learning to make our AlphaGo model mimic all models, in order to achieve artificial general intelligence (AGI)
We are very sure that this will work out as we are using the latest smartphone to make calculations. Also, we are ready to invest in buying another smartphone to exploit the wisdom of crowds.

Link: https://github.com/skyfallen/AGI (this is an example)

Analyzing audio data with DTW algorithm
Team: Gulnar Mammadli, Altay Garayev
This project aims to analyze 2 different audios and find out similarities between them with the help of DTW. We will be using many different audio files for this and will visualize the results.
Link: https://github.com/Gulnar-Mammadli/dtw-audio

Title: SUDOKU solver
Team: Farida Rustamova, Leyla Rahimli

SUDOKU is the one of the popular games all over the world. This project aims to create an algorithm to solve 9X9 sudoku. The algorithm we created will be able to solve sudoku regardless of the difficulty level and visualize it.

Link: https://github.com/faridarustamova/Sudoku.git

Genetic algorithms comparison with A* for finding shortest path in network
Team: Araz Heydarov

A* is one of the good and basic tools for finding the shortest path between point A and B in a network. Applying genetic algorithm for finding the path may improve time complexity of this problem in some cases.

Link:

Generating Simple Musical Compositions Using Genetic Algorithms
Team: Anton Slavin

The main objective and idea of this project is to explore different ways of creating genetic algorithms capable of writing musical pieces such that they would subjectively be "not horrible", as scored by humans. The main challenge will be to create a proper fitness evaluation function which will clearly define the boundaries between a horrendous piece of music and a passable one.

Link: https://github.com/tonysln/music-gen

Step-by-step visualization of maze generation algorithms
Team: Kermo Saarse, Taras Ivantsiv

This is an interactive application where the user can select the size of the maze and an algorithm and the application would visualize each step of the maze generation. Possible algorithms would be recursive backtracker, randomized Kruskal's algorithm, randomized Prim's algorithm and Wilson's algorithm. I got inspiration for this project from here: https://weblog.jamisbuck.org/2011/1/20/maze-generation-wilson-s-algorithm

Demo: https://ivantsiv-ut.github.io/
Code: https://github.com/kermo-git/MazeAlgorithms

Space Partitioning for Video Game Rendering by Using Graphs
Team: Mert Bektas, Alicia Sudlerd, Gandab Hasanova

Binary space partitioning is a method that takes the environment in video games and splits it into parts, similar to a KD-tree, and creates a Binary Space Partitioning (BSP) tree out of it. Binary space partitioning was developed in the context of 3D computer graphics and used in many video games, such as "Doom," which is one of the most significant games in video game history, to increase games' rendering performance. We aim to implement an algorithm like this, using graphs instead of BSP trees. Additionally, analyze it to see if graphs would make this algorithm faster.

Link: https://github.com/Alitcher/Algorithmics-BSPMapGenerator

---

Using shortest path finding algorithms on real-world data
Team: Kyrylo Riazantsev, Ralf Tambets

Employ Dijkstra and A* with different search heuristics on a roadmap of Tartu to find the shortest paths between locations of interest. Compare the routes obtained with what Google Maps suggests.

Link: https://github.com/ralf-tambets/algorithmics_project

3D maze solving with a genetic algorithm
Team: Tiit Vaino, Jakob Univer, Sander Roosalu

Our goal is to find an optimal solution for a ball thrown into a 3-dimensional maze. Goal of the ball is to arrive at a designated spot in the maze. For finding the optimal solution, we are trying to use a genetic algorithm. The maze can consist of walls with different effects on the ball etc.

Link: https://github.com/JakobUniver/algorithmics_3D_maze

ENTROPY solver
Team: Daniel Würsch, Herman Rull

ENTROPY is a turn-based board game, where players insert tokens of different colors to the board. One player aims to create order through patterns, while the other player tries to increase entropy by avoiding patterns.
The final solution is built to satisfy the restrictions of CodeCup, so that we could compete in the competition. https://www.codecup.nl/entropy/rules.php

Link: https://github.com/danwue/entropy

Solving Boggle with different initial conditions and analysing the solutions
Team: Martin Vainikko

Boggle is a word game where the players are given a 4x4 (or a 5x5) grid consisting of randomly arranged dice with letters on each side. The idea is to find as many words as possible arranging the letters on the top side of the dice by connecting (including diagonally) adjacent dice. The goal of the project is to implement the game and a solver to then analyse the variation of solutions on different initial setups of the game: different dice (maybe comparing different languages) and different random dice placements.

Link: https://github.com/nitram-v/boggle-project

Title: Text compression algorithms: Huffman coding, LZ77 and Burrows-Wheeler transform.
Team: Erki Külaots, Kadi Sammul, Taaniel Saarnik

Implementing and comparing text compression algorithms: Huffman coding and LZ77 with and without doing Burrows-Wheeler transform.

Github: https://github.com/Sabasik/Algoritmika

Title: Implementing flow field pathfinding and comparing it to A*.
Team: Robert Allik

A* can get quite slow when there are many units calculating their paths especially if the world is dynamic. Flow field pathfinding is a more efficient way to find correct paths for all units. The objective is to make a GUI program that implements both A* and flow field, then compare the algorithms in different conditions.

Github: https://github.com/SuspiciousSeal/a_star_vs_FF

Comparing game theory algorithms in Poddavki
Team: Karl-Johan Pilve, Kaspar Kadalipp, Eduard Rudi

Comparing game theory algorithms speed and effectiveness in Poddavki. Minmax, alpha-beta pruning and Monte Carlo tree search - these are the algorithms that we are using in our project. The game Poddavki is like Checkers, but the goal is opposite, the winner is, who loses all its pieces first. We also describe which of the following algorithms is the easiest to implement.

Github: https://github.com/EduardNot/Poddavki-AI-algorithms

A* Obfuscator
Team: Glib Manaiev, Dmytro Fedorenko

Using an A* algorithm to process images and, as a result create a new pixel-art images. Additionally to the program itself, there will be a Telegram-bot, allowing users to use the algorithm on their own images.

Github: Dimonitr/Astar_obfuscator_algorithmiccs: Algorithmics (MTAT.03.238) project by Glib Manaiev and Dmytro Fedorenko (github.com)

Project Topics (from Previous Years)

Developing a Skip-list-inspired data structure with O(n) space complexity
Team: Bruno Rucy, Joonas Järve and Mykyta Baliesnyi
The skip list has three main characteristics; skipping, height hierarchy and ease of concurrency, however it has a mortal flaw, O(n log n) space complexity. Can we somehow adapt it to O(n) and still retain at least one of its characteristics? Is it possible to skip(literally because the skip list doesn't really skip it just goes sideways or down) some of the heights during the traversal? use min and/or max values? Try out the idea from the Splay-lists paper and make height a factor of the access distribution?. The main source of inspiration came up when attempting to represent skip lists as a directed hypergraph interpretation of multisets.

Link: the github repo that we'll create

Image Reproducing with Genetic Algorithm
Team: Ivan Hladkyi, Denis Kolomyets, Tejas Shah
We are creating a program that uses genetic algorithm to reproduce the input image with simple geometric shapes, e.g., lines, circles, triangles.

Link: https://github.com/HladkyiIvan/GA_ImageReproducing

Comparing parallel sorting algorithms and their non parallel versions
Team: Silver Maala, Cathy Toomast, Aleksander Nikolajev

We are implementing parallel sorting algorithms and their non parallel versions. Afterwards we benchmark them on various datasets and compare their results. We will compare the parallel algorithms with each other to find the best one and we will also compare the parallel sorting algorithms to their non-parallel versions.

Link: the github repo that we'll create Wanna compare CT

Trading with dynamic programming
Team: Karl Kaspar Haavel, Markus Haug

Assuming a world where momentum strategies work and denying the efficient market theory, the project will abuse "repeating" patterns in markets for profits. The goal is to create and compare trading algos using dynamic programming with different distance metrics in order to find the ones that can reliably beat the market. Distance metrics that will be looked into: dynamic time warping, longest common subsequence and time edit warp distance.

Link: https://github.com/ka5par/dtw

Hyperparameters tuning for machine learning models using evolutionary algorithms.
Team: Farid Hasanov, Farid Valiyev, Fatima Gurbanova.

Usually, search for optimal hyperparameters (that lead to better score) is done at the end of the ML pipeline. It is done using Grid Search, or Random Search algorithms (there are also others, but these as I know are the most popular). We want to try out several evolutionary optimization algorithms in the problem of search for the most optimal set of hyperparameters. We will evaluate it with different supervised machine learning models - primarily with those that need heavier hyperparameter tuning (to be determined). Proper cross validation schemes will be built and shown.

Link : github repo to be added later

Algorithmic generation of string-art
Team: Kert Tali; Kaspar Valk

String-art is a type of handicraft, in which fine thread is pulled through hooks on a canvas, creating the illusion of depth and shapes, which forms an image (eg. a portrait of a person). An algorithm which calculates the order of hooks through which thread has to be pulled is the only way for an artist to create such a piece. Our objective is to create and refine such a tool.
Link: Github

Reinforcement learning using evolving agents
Team: Marti Ingmar Liibert, Carel Kuusk

Learning to play one of the games from OpenAI Gym using deep reinforcement learning and genetic algorithms to evolve the agents.

Link: https://github.com/ingmarliibert/evolve-rl

---/---

Analyzing audio data with DTW algorithm
Team: Hashim Hashimov, Kayahan Kaya

This project will detect similarities between two different audio signals using Dynamic Time Warping algorithm and calculating the rate of similarities. We will test our algorithm with different types of audio signals to find out the best type of audio signal and visualize it.

Link: TBD - Github

Genetic Algorithms for Evolutionary Art
Team: Rodion Krjutškov, Polina Shevchenko

We are making a small web application for generating Suprematism artworks and improving them with genetic algorithms based on user input.

Link: TBD

Applications of DTW in music
Team: Alfred Saidlo, Andreas Pung, Thamara Luup

Use DTW to match different audio files. Visualisation.

Link: TBD - Github

Dynamic Programming Study Tool (DPST)
Team: Bohdan Romashchenko, Artem Grukhal, Jatan Shrestha

Provide an online tool which would help students to understand Dynamic Programming. It should have useful theory and an ability to make visualizations.
Link: Github

Implementing the BLAKE-3 algorithm
Team: Karl Hannes Veskus

BLAKE-3 is a new (2020) cryptographic hash function that is designed to be as fast as possible, easy to use and implement, while not losing in security. Implementing a cryptographic hash properly is non-trivial as the algorithm works at a very low-level and the security properties should still hold. This means that any language peculiarities and differences from the source algorithm have to be thought through and implemented with precision. Additionally, as the function is recent and state-of-art, it is reasonable to assume that even understanding the algorithm might present challenges.

Link: GitHub

Flood-It

Team: Kristiine Saarmann, Ulrike Engeln

Program the game Flood It with some special tweaks. Apply the Greedy algorithm to calculate the upper bound to the number of steps taken by the player. Find an optimal solution for the game, maximizing the points with the upper limit to the moves.

Link: TBA

Real-time document search using Radix Trees
Team: Juan Carlos Ramos, Sergei Tsimbalist

Substring matching is usually constrained to one document and one result at a time. We improve this by implementing a real-time substring matching on multiple documents at the same time using the power of Radix Trees.

Link: TBA

AVL-tree simulator
Team: Mona Küüts, Timo Tiirats

Our aim is to create a GUI that will help students to understand how an AVL-tree works. It should allow the students to change the parameters of the tree and see what happens accordingly.

Link: https://github.com/MonaKyyts/algode_projekt

Analysis of maze generation and path finding algorithms
Team: Hanna Britt Parman, Kristiina Keps

Our goal is to implement a few maze generation algorithms, compare them (the output and time) and try out different path finding algorithms (both maze path finding and general shortest path algorithms), visualize their work and compare the efficiencies.

Link: https://github.com/kristiinakeps/pathfinding

Analysis of Delta data and DTW
Team: Kairit Peekman, Anisa Bruçi, Kaire Koljal
Analyzing data collected by different sensors in Delta and creating time series.

Link: https://gitlab.com/mscprojects1/deltadata

Motif discovery in COVID-19 related data
Team: Karmen Kink and Ainika Adamson

The goal is to identify similar patterns to Estonian COVID-19 related data from other countries to help predict what may await Estonia in the future.
Link: TBA

Shortest Path Visualisation using A* algorithm
Team: Fidan Rustambayli and Shekhar Suman Patel
Our goal is to create a GUI app that visualize A* algorithm. There will be destination point and students will be able to draw walls around it on the grid and start the game then A* algorithm will find the destination point without crossing the walls.

Link: https://github.com/Fidanrlee/Shortest-Path-Visualisation-using-A-algorithm.git

---

Quantum tunneling
Team: Anabel Ovide, Handy Kurniawan, Alejandro Villoria
Quantum tunneling is a quantum physical phenomena without a classical analogy. It can be compared to Simulated Annealing in the way it can be also used to solve optimization problems. It refers to situations where quantum systems can be found in superposition between states separated by a high and narrow energy barrier. Quantum tunneling can be exploited by optimization. The objective is to explain and visualize quantum tunneling by computing simulation.
Link: https://github.com/alejandrovgonzalez/quantumAnnealing

Drone navigation using A*
Team: Agnes Luhtaru, Oliver Vainumäe

Having done a project in Machine Learning using drones, we were interested in utilizing the drones again for this course.

- In progress

Link: TBA

Evolutionary creature generation
Team: Sander Sats; Tetiana Rabiichuk, Anti Kivi

The main idea of the project is to have a world with creatures that eat and multiply (possibly something more if we have time) and have different attributes (speed, size etc) which are controlled by genetic information which gets changed by mutations and crossover on multiplication.
The objective is to see what will be the evolutionarily stable outcome.

Link: https://github.com/liivur/algorithmics-project

Benchmarking GPU and CPU parallel sorting algorithms

Team: Rasul Nabiyev

When we talk about parallel processing and concurrent algorithms in recent years Graphic Processing Units excels traditional CPUs with orders of magnitude more core counts. NVIDIA's open source CUDA toolkit allows developers to utilize the power of their GPUs with convenient(arguably) C++ APIs. The aim of this project is to implement and test some parallel sorting algorithms on GPU and compare them with their CPU counterparts.

Link: https://gitlab.cs.ut.ee/nabiyev/gpu-sorting-benchmark

Title: Y chromosome haplogroup identification using breadth-first search and array comparisons

Team**:** J. Rodrigo Flores E.

Current human Y chromosome haplogroup assignment methods work on the basis of performing a breadth-first search starting at the root of the known Y phylogeny and for every descending branch from a parent node the algorithm moves to the next node for which there exists more evidence. This generally works well for samples for which the state of many positions is known. However for ancient DNA samples the total positions of known states is rather limited, and almost invariably the previous strategy will stop at an early node near the root due to lack of information. Here I implement an alternative strategy that also does a breadth-first search but coupled with a basal complete linear exploration on all informative positions using reference predictor panels. This strategy guarantees a best possible estimate even in a context of limited data.

Link: TBA

Title: Applying Path finding algorithm on real world Map

Team: Shumpei Morimoto, Tazo Sepiashvili, Manish Gupta

The objective of the project is to traverse a real world map image by applying pathfinding through the road on the image. The goal is to traverse through multiple points on the map only via Road, and attempt to implement TSP by selecting the shortest route in doing so. This algorithm can be used as a route optimizer for someone who needs to visit multiple locations in a real map.

Link: https://github.com/manishgupta94/finding-path

Title: Military Path Finder using path finding algorithms (Military.GPS)

Team: Risto Künnapas, Friedrich Krull

The need to traverse landscape as quickly as possible is very important when conducting warfare, but only traversing the shortcuts/roads could be very dangerous, so it may be necessary to look for paths in forests. Here we try and implement shortest path finding

algorithms that also take into account the safety of the troops and thus finding the quickest yet safest route.

Link: https://github.com/ristokynnapas/algorithmics_project

Title: Using a genetic algorithm as a gameplay mechanic to play Chrome Dino game🦖
Team: Prakhar Srivastava, Mohga Emam, Aleksandr Krylov

In the game, we implemented our own genetic algorithm which mimics the process of natural evolution that is selection, mutation and crossover. In the framework, the population of Dino evolves iteratively in each generation to increase the fitness of Dino to jump over all the cacti.

Git link: https://github.com/prakharsdev/ChromeDino

Title: Perceptual hashing
Team: Georg Reintam, Herman Klas Ratas

Uploading photos to the internet could be considered one of the most popular activities of the 21. Century, which means that large amounts of data is added to the internet. Therefore, high volume of data needs to be scoured to verify the suitability of media for the large audience. Different computer science techniques, like perceptual hashing, help to navigate the huge haystack more quickly and efficiently.

Git link: TBA

Title: Mutating Polynomials for Image Approximation
Team: Ott-Kaarel Martens, Joonas Praks

The idea is to approximate images with shapes defined by polynomial functions. The polynomials are iteratively mutated with a simple genetic algorithm, to approximate the base image.

Github: https://github.com/ottmartens/approximate-landscapes

Title: Evolution Algorithms For ASCII Art
Team: Donatas Vaiciukevičius, Dzvenymyra-Marta Yarish, Nikita Fordui

The project is based on a similar idea where it was attempted to approximate images using triangles. Here the idea is to take an existing image, and convert it to as close of a representation to the original as possible, the output should be a text file and/or image with the generated image representation.

Git repository: https://github.com/Donce530/ASCII-art-by-evolutionary-algorithms

---

Title: Grayscale imaging with a single line
Team: Raido Everest

One of the example ideas, actually - the idea is to take an image, essentially turn it into a graph, and use TSP to find some path through the darker areas of the image and create a visualization. Also a somewhat comfortable interface.

Github: soon

Title: Auto Rubik's Cube Solver
Team: Joshua Katigbak, Rasmus Lellep, Yuliia Shevchenko

The goal of our project is to implement the Algorithm's proposed in 'Algorithms for Solving Rubik's Cubes' by Demaine E., et al. Ideally with a nice UI/animation that shows the process the algorithm goes through in solving the Rubik's cube.

Link: TBD

Title: Optimization for fitting the best curve using the differential evolution
Team: Illia Tsiporenko, Dmytro Shvetsov

Using the data generator, we create a polynomial function and the data points based on this function. The task is to guess the polynomial coefficients using Differential Evolution and implementing the user friendly interface to play around with it.

Github: https://github.com/Dmytro-Shvetsov/de-poly-approx.

Title: Distributed Hash Table in Peer-to-peer systems using Chord algorithm
Team: Amageldy Shalginbayev, Dariya Nagashibayeva

The project work will involve the research and development of the distributed hash tables applied in peer-to-peer systems. Our practical work will be mostly based on the paper Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications
https://pdos.csail.mit.edu/papers/ton:chord/paper-ton.pdf

Github: https://github.com/amangeldyshalginbayev/ChordProtocol

---

Title: Pocket Cube: Work less or faster?
Team: Joosep Hook

The goal of the project is to explore different ways to speed up both enumerating all possible cube configurations and solving any cube configuration using BFS. But if we want to speed up our solver, what yields bigger payoffs: doing less work by only visiting unique cube configurations and avoiding taking duplicate paths, or sticking to the naive BFS algorithm and implementing more efficient elementary operations?
Link: https://github.com/joosephook/pocket-cube