Algorithms Projects'22

Requirements Summary

**Group Count:** 3 members

**Group Registration:**

* [**Online form**](https://docs.google.com/forms/d/e/1FAIpQLSdhSLHMQJexx7Ox5wLxMWdxwwUlO1zpu9lwTJpBBi3cHo_1MQ/viewform) due to **THR 21-APRIL-2022**, (**After deadline:** Groups will be **Manually Assigned** to a Project)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project** | **Allowed CODE** | **Inputs & Outputs** | **Deliverables** | **Grades[[1]](#footnote-2)** | **Bonus** |
| [**Intelligent Scissors**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b1%5d%20Intelligent%20Scissors/Intelligent%20Scissors.docx) | 1. Open and display the images. 2. Function to calculate the edge-strength G between two pixels. **(Check the** [**document**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b1%5d%20Intelligent%20Scissors) | **Inputs:**   1. Image (2D array of pixels). 2. Anchor point. 3. Free point.   **Outputs:**   1. Path between the anchor and the free point. 2. Final lasso closed path. | **Document contains ONLY:**   1. Graph construction description and code. 2. Used shortest path algorithm code. 3. Detailed analysis of the above codes. | 40% | 1. As you can see in [Photoshop example](https://www.youtube.com/watch?v=n6nHAnc0E2E&feature=youtu.be) you can 1) Click to place anchor. 2) Move the mouse to generate the livewire. 3) When the wire’s length exceeds a certain length, an automatic anchor point is placed to make the wire more stable. Implement a similar algorithm that automatically places new anchor points. 2. Add the ability to increase the frequency of anchor points in some critical regions. |
| **Implementation**:   1. Graph construction. 2. Shortest path EFFICIENT implementation. 3. Path backtracking and drawing. 4. Support multiple anchor points. 5. Generating a closed lasso. | 60% |
| [**Image Quantization**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b2%5d%20Image%20Quantization/Image%20Quantization.docx) | **C# TEMPLATE Code** to   1. Open image & load it in 2D array 2. Calculate the edge-strength G between two pixels 3. Display image.   **PRIORTY QUEUE Code** | **Inputs:**   1. Image (2D array of pixels). 2. Num of clusters (K).   **Outputs:**   1. Quantized image 2. Color palette | **Document contains ONLY:**   1. Graph construction description and code. 2. Minimum spanning tree code. 3. Palette generation code. 4. Detailed analysis of the above codes. | 40% | * 1. Automatic Detection of the Number of Clusters   2. Better Way for Finding the (K) Clusters |
| **Implementation**:   1. Graph construction. 2. EFFICIENT implementation of minimum spanning tree. 3. Extracting the K clusters. 4. Palette generation by calculating the clusters centroids. 5. Mapping the original colors to the palette colors. | 60% |
| [**N PUZZLE**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b3%5d%20N%20Puzzle/N-Puzzle.docx) | **PRIORTY QUEUE Code** | **Inputs:**  **File contains:**   1. ***N:*** puzzle size 2. Initial state (rows and cols)   **Outputs:**   1. Solvable or not. 2. Min number of moves to arrange the puzzle. 3. Display board at each move | **Document contains ONLY:**   1. Entire Source code. 2. Detailed analysis of your code. 3. Hamming vs. Manhattan over "**Complete Test**" showing: Min number of moves & Execution time | 40% | 1. Apply **different shortest path algorithm** (BFS or DFS) and compare it with A\* 2. **User friendly GUI:** to rewind the search one step at a time over a generic N-puzzle. (e.g. NxN matrix of picture boxes & swap). |
| **Implementation**:   1. **Read** and parse the initial state file 2. **Determine** if it is solvable or not? 3. R**epresent** the N-puzzle in form of tree 4. **Implement A\*** with Hamming & Manhattan 5. **Print a STEP by STEP** movements | 60% |
| [**Small World Phenomenon**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b4%5d%20Small%20World%20Phenomenon/Small%20World%20Phenomenon.docx) | **NONE** | **Inputs:**   1. **Movies file:** Each line has 2. a movie title, 3. list of actors/actresses delimited by '/'. 4. **Queries file:** Each line 5. pair actors/actresses separated by '/'.   **Outputs:**  For each query pair, print out:   1. degree of separation 2. relation strength 3. shortest chain of movies | **Document contains ONLY:**   1. Entire source code 2. Detailed analysis of your code   Execution time of "**Complete Test**" cases: before and after applying the optimization idea | 40% | * 1. Calculate and print the **distribution** of degree of separation between a given actor and all others.   2. Find and display the **strongest path** (based on the relation strength) between a query pair of actors.   3. Find & display the **min number of movies** that link all actors together. |
| **Implementation**:   1. **Read** and parse two files (Movies & Queries) 2. For each pair, **print out** the following: 3. **- degree of separation** 4. **- relation strength** 5. **- shortest chain of movies** 6. **Optimize** the calculation of the degree of separation to be run in ***sublinear*** time | 60% |
| [**WordNet Semantics**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b5%5d%20WordNet%20Semantics/WordNet%20Noun%20Semantics.docx) | NONE | **Inputs:**   1. List of synsets 2. List of hypernyms (parents) 3. Semantic relations queries 4. Outcast detection queries   **Outputs:**   1. Semantic relatedness (dist, sca)   Detect the outcast noun | **Document contains ONLY:**   1. Describe the data structure(s) you used to store the information of “synsets” file. Why did you make this choice? 2. Describe the data structure(s) you used to store the information of “hypernyms” file. Why did you make this choice?   Describe your algorithm to compute the shortest common ancestor. Show the detailed analysis of the corresponding code? | 40% | 1. More efficient solution of SCA between Two Synsets IDs 2. Efficient Calculation of Distance and SCA between Two Nouns |
| **Implementation**:   1. Graph construction 2. Two mapping functions: 3. Noun to SynsetsIDs 4. SynsetsID to Nouns 5. Distance and the shortest common ancestor between: 6. Two synsets IDs 7. Two words 8. Answer the two questions by finding: 9. Semantic relatedness   Outcast noun | 60% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Delivery** | [**Intelligent**](%5b2%5d%20Intelligent%20Scissors/%5bALG'18%5d%20Intelligent%20Scissors.docx)[**Scissors**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b1%5d%20Intelligent%20Scissors) | [**Image Quantization**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b2%5d%20Image%20Quantization/Image%20Quantization.docx) | [**N PUZZLE**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b3%5d%20N%20Puzzle/N-Puzzle.docx) | [**Small World Phenomenon**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b4%5d%20Small%20World%20Phenomenon/Small%20World%20Phenomenon.docx) | [**WordNet Semantics**](https://cisasuedu-my.sharepoint.com/personal/ghadahamed_cis_asu_edu_eg/Documents/%5bALG'22%5d%20Materials/Staff%20Only/%5b10%5d%20Project/Release/%5b5%5d%20WordNet%20Semantics/WordNet%20Noun%20Semantics.docx) |
| **Milestone1**  **(START of Week Before Practical Exam)** | 1. Construct a weighted graph for an image 2. EFFICIENT implementation of shortest path between start anchor point and free points. 3. Backtrack the shortest path from a free point (mouse position) to the anchor point 4. Draw the path on the image by supporting multiple anchor points 5. Documentation | 1. Construct a weighted graph for distinct colors in the image 2. Implementing the minimum spanning tree of the graph. 3. Extracting the K clusters from the minimum spanning tree. 4. Calculating their centroids (Palette Generation) 5. Quantize the image by replacing the colors of each cluster by its representative color. 6. Documentation | 1. **Read** and parse the input file 2. R**epresent** the N-puzzle in form of graph/tree 3. **Implement A\*** search algorithm using **Hamming Distance** 4. STEP by STEP display for the 3x3 case 5. **Modify A\*** search algorithm to use **Manhattan Distance** 6. Find a formula to indicate whether any given game state is solvable or not. 7. Documentation | 1. **Read** and parse two files (Movies & Queries) 2. Calculate **degree of separation** for query pair 3. Calculate **relation strength** for the query pair 4. Print **shortest chain** of movies 5. Apply **optimization** idea 6. Documentation | 1. **Graph** construction 2. **Two mapping functions**    1. Noun to SynsetsIDs    2. SynsetsID to Nouns 3. **Shortest common ancestor** between two synsets IDs 4. Two synsets IDs 5. Two words 6. **Answer** the two questions by finding:    * + Semantic relatedness      + Outcast noun 7. Documentation |
| **For Milestone1:**   * + **MUST** deliver the required tasks and **ENSURE** it’s worked correctly   + **MUST** deliver in your scheduled time (TO BE ANNOUNCED) | | | | | |

1. Grades distribution is subject to change without prior announcement [↑](#footnote-ref-2)