Algorithms Projects'16

Requirements Summary

**Group Count:** 3 – 5 members

**Group Registration:** [**Online form**](https://docs.google.com/forms/d/e/1FAIpQLSfShoDMsGLtdUJuAPnd5sCCnAA2GawDvymYlcADC69343Qq3g/viewform) due to **SAT 26-Nov-2016**, (**After deadline:** Groups will be **Manually Assigned** to a Project)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project** | **Allowed** | **Inputs & Outputs** | **Deliverables** | **Grades[[1]](#footnote-1)** | **Bonus** |
| [**Image Quantization**](%5b1%5d%20Image%20Quantization/Image%20Quantaization.docx) | **C# Code** to   1. Open image & load it in 2D array 2. Calculate the edge-strength G between two pixels 3. Display image. | **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% |
| [**WordNet Semantics**](%5b2%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) 2. 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? 3. 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 10. Outcast noun | 60% |
| [**Autocomplete Me**](%5b3%5d%20Autocomplete%20Me/Autocomplete%20Me.docx) | NONE | **Inputs:**   1. English dictionary file. 2. Search sentences file. 3. User input query via GUI   **Outputs:**   1. GUI based app 2. Top matched results sorted by their weights. 3. Suggestions list of the most nearest miss-spelled words | **Document contains ONLY:**   1. Algorithms description and code of each stage of the flow chart    1. Prefix Search    2. Sort    3. Suggestion    4. Substring Search. 2. Detailed analysis of the above codes. | 40% | 1. Efficient substring searching. 2. Don’t search again when user completes typing. 3. Any other efficient recommendations are welcomed. |
| **Implementation:**   1. Create GUI for the user. 2. Efficient prefix search. 3. Compare between two different “Complexity” sorting algorithms for weights- sorting. 4. Efficient algorithm to check whether a word is correctly spelled or not. 5. Efficient suggestion algorithm. 6. Substring search algorithm. 7. Display the list of suggestions and results for the user. | 60% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Delivery** | [**Image Quantization**](%5b1%5d%20Image%20Quantization/Image%20Quantaization.docx) | [**WordNet Semantics**](%5b2%5d%20WordNet%20Semantics/WordNet%20Noun%20Semantics.docx) | [**Autocomplete Me**](%5b3%5d%20Autocomplete%20Me/Autocomplete%20Me.docx) |
| **Milestone1**  Week12: start at SAT 10-12-2016  During LABS | 1. Construct a weighted graph for distinct colors in the image 2. Greedy algorithm of single linkage clustering based on implementing the minimum spanning tree of the graph. 3. Documentation I | 1. Graph construction 2. Two mapping functions 3. Shortest common ancestor between two synsets IDs 4. Documentation I | 1. Create GUI for the user. 2. Efficient prefix Search. 3. Use and compare between two different “Complexity” sorting algorithms for weights. 4. Display the list of results. 5. Efficient algorithm to check if a word is correctly spelled. 6. Documentation I |
| **Milestone2**  Final Delivery  Lab Exam week start at SAT 31-12-2016 | 1. Extracting the K clusters from the minimum spanning tree. 2. Calculating their centroids (Palette Generation) 3. Quantize the image by replacing the colors of each cluster by its representative color. 4. Documentation II | 1. Shortest common ancestor between two nouns 2. Semantic relatedness between two nouns (distance and shortest common ancestor) 3. Outcast noun in a given list of nouns 4. Documentation II | 1. Efficient suggestion algorithm. 2. Substring searching algorithm. 3. Display the list of suggestions. 4. Documentation II |
| **For Milestone1:**   * + **MUST** deliver the required tasks and **ENSURE** it’s worked correctly   + **MUST** deliver the **part of the documentation** that is related to the Milestone (printed document)   + **MUST** deliver in the **section time of the majority members** (e.g. if a group consists of 3 members from sec.1 and two members from sec.2, they should deliver in the time of sec.1) | | | |

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