*check tonight for links to 2 example projects

CSCI 432 Project

Overview

In this project, you will work in a group (of 3-4 students) to create a video to present one of the following two options:

1. A recent algorithm +1. Here, you will present an algorithm from a "recent" conference or journal (note: recent means within the last 15ish years). You will choose the algorithm on your own, and become experts on this algorithm. In addition, you will take the paper that they found one step further in the plus-one (+1) element of their project.

2. Comparison of algorithms. Here, you will choose a problem with at least three algorithms (and/or impmenetations) to compare. You must implement at least one of these algorithms. The presentation is to present the pros/cons of the different algorithms/implementations.

Deliverables

Although not mandatory, all teams are encouraged to meet with the instructor or the TA.

For you!

- 29 November 2021: Progress report due. In an up to three page document, explain what your group has accomplished so far, what the immediate next steps are, and what the course of action is to complete the final project. By the time you submit this deliverable, it is expected that you will have communicated with the professor and/or TA either via email or in person or via Zoom.

last week of class

- 6 December 2021: final presentations. The presentation will be in the form of a 5-minute video (min: 4 minutes, max: 6.5 minutes). After presenting the video in class, be prepared for a short Q&A. HERE
- 10 December 2021: A formal write-up summarizing the algorithm that your group presente is due, including a description of the problem it solves. In addition,
  1. For those that choose the "recent algorithm +1" option, describe your +1 option in detail.
  2. For those that choose the "comparison of algorithms" option, please fully describe each algorithm/implementation, as well as the comparison techniques.
- FINAL: Please submit a 2-3 page document outlining each group member's contribution to your group. This is an individual submission. INDIVIDUAL submission → can include your contributions to discussions on the LWOE
- Please submit all write-ups in both D2L and Gradescope. For any code written for this project, please submit via D2L only.

Option 1 Suggestions

List of possible papers from which to base a project:

- Ahmed, Fasy, Wenk. Path-Based Distance for Street Map Comparison. ACM TSAS, 2015.
- Aloupis, Fevens, Langerman, Matsui, Mesa, Nu˜nez, Rappaport, and Toussaint. Algorithms for Computing Geometric Measures of Melodic Similarity. Computer Music Journal, 2006.
- Bethea and Reiter. Data Structures with Unpredictable Timing. ESORICS, 2009.
- Chambers, de Verdire, Erickson, Lazard, Lazarus, and Thite. Homotopic Frchet Distance between Curves or, Walking your Dog in the Woods in Polynomial Time. Computational Geometry, 2010.
- Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. Discrete and Computational Geometry, 1996.
- Chen and Kerber. Persistent Homology Computation with a Twist. EuroCG, 2011.
- Chazal, Fasy, Lecci, Rinaldo, Singh, and Wasserman. On the Bootstrap for Persistence Diagrams and Landscapes. Modeling and Analysis of Information Systems, 2013.
- Edelsbrunner and Pausinger. Approximation and Convergence of the Intrinsic Volume. Adv. Math. 287

(2016), 674-703

- Fajstrup, Goubault, Haucourt, Mimram, and Raussen. Trace spaces: An Efficient New Technique for State-space Reduction. ESOP, 2012.
- Guibas, and Oudot. Reconstruction Using Witness Complexes. DCG, 2008.
- Li and Svensson. Approximating k-Median via Pseudo-Approximation. STOC, 2013.
- Liota. Low Degree Algorithms for Computing and Checking Gabriel Graphs, Brown University Technical Report, 1996.
- Miller and Sheehy. Approximate Centerpoints with Proofs. CGTA, 2010.
- Millman, Love, Chan, and Snoeyink. Computing the Nearest Neighbor Transform Exactly with only Double Precision. ISVD, 2012.
- Milosavljevic, Morozov, and Skraba. Zigzag Persistent Homology in Matrix Multiplication Time. SoCG, 2011.
- Pratt, Riley, and Sheehy. Exploring Circle Packing Algorithms. SoCG, 2016.
- Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing, 1997.
- Oostrum and Veltkamp. Parametric Search Made Practical. SoCG 2002.
- Wang, Wang, and Li. Efficient Map Reconstruction and Augmentation via Topological Methods. ACM SIGSPATIAL GIS, 2015.
- Zheng, Jestes, Phillips, Li. Quality and Efficiency in Kernel Density Estimates for Large Data. ACM Conference on the Management of Data (SIGMOD), 2013.

The list of papers above is only a sprinkling of recent developments in algorithms. I encourage teams to peruse the following (non-exhaustive) list of venue for finding other papers:

- ESA: European Symposium on Algorithms.
- EuroCG: European Workshop on Computational Geometry.
- GD: International Symposium on Graph Drawing.
- ISAAC: International Symposiums on Algorithms and Computation.
- LATIN: Latin American Theoretical INformatics.
- SIGSPATIAL GIS
- SoCG; Symposium on Computational Geometry.
- SODA: Symposium on Discrete Algorithms.
- STOC: ACM Symposium on Theory of Computing.
- SWAT: Scandinavian Workshop on Algorithm Theory.
- WADS: Algorithms and Data Structures Symposium.

Also check out algorithm competitions, such as the Geometric Optimization Challenge.

22 Oct 2021

Whatever First Search

→ goal: to traverse through a graph from a given start node.

~~Count conn. components~~

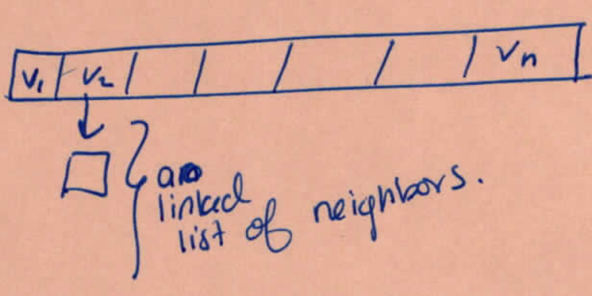
~~J = 1 (global variable we need now)~~

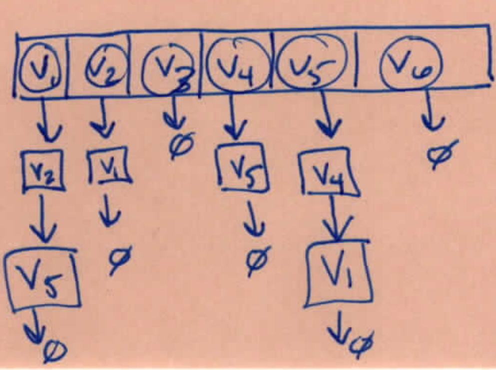
```
DFS (G, n)
    mark vertex n as visited
    for all children c of n
        DFS (G, c)
    dosomething (n)
    end for
```

(rep. as circling below) if dosomething(n) were up here, that would be a preorder traversal.

postorder traversal

Given a graph G, we represent it as an adjacency list

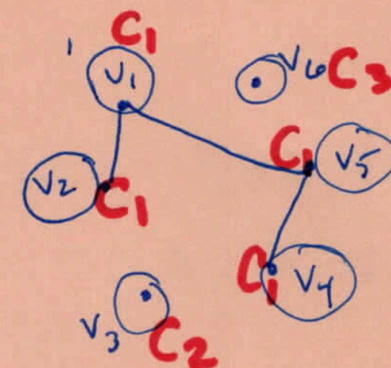




Q How many connected components are there?

dosomething(n) := mark v_n as in component J

global varia



```
Count Conn Comp (G)
  J = 1   make global variable
  for v_1, ..., v_n
      if v_i not yet marked
          DFS(G, i)
          J++
      end if
  end for
```

Time Complexity: $\Theta(n+m)$

caution: back of hand calc will tell you $O(\underset{\uparrow \text{ for loop}}{n}(\underbrace{n+m}_{\text{DFS}}))$

$a, b \in \mathbb{R}^n$ \preceq

The product order $a \preceq b$

iff $a_i \leq b_i \quad \forall i \in \underline{n} := 1, 2, \ldots, n$

$S \subseteq \mathbb{R}^n$, we can find a total order compatable w/ the product order.

$\forall a, b \in S$

if $a \preceq b$, then a comes before b