Curriculum Graph Visualizer App

SUMMARY

THE GRAPH OF COURSES AND THEIR PREREQUISITES IS COMPLEX, AND STUDENTS AND FACULTY CAN BOTH STRUGGLE TO VISUALIZE IT. A POTENTIAL SOLUTION IS TO MODEL THE CURRICULUM AS A DIRECTED GRAPH, WHERE EACH COURSE IS A VERTEX AND EACH PREREQUISITE REQUIREMENT IS A DIRECTED EDGE. IF THIS GRAPH IS SERIALIZED IN A DIGITAL FORMAT SUCH AS DOT OR ONE OF THE FORMATS SUPPORTED BY NetworkX, THEN IT IS POSSIBLE TO WRITE PROGRAMS TO VISUALIZE THE CURRICULUM, SIMULATE STUDENT PROGRESS THROUGH THEIR STUDY PLANS OR THE CURRICULUM ITSELF. THIS INVOLVES FUNDAMENTAL GRAPH ALGORITHMS, GRAPH DRAWINGS, HUMAN COMPUTER INTERFACES, AND POTENTIALLY AI PLANNING ALGORITHMS.

PREREQUISITE READING

- ✓ THE GRAPH IN GRAD HANDBOOK
- ✓ CURRICULUM DEVELOPMENT USING GRAPHS OF LEARNING OUTCOMES
- ✓ DOT GRAPH FORMAT
- ✓ GRAPH FORMATS SUPPORTED BY NetworkX
- ✓ GEOMETRY IN ACTION: GRAPH DRAWING

ADVISORS

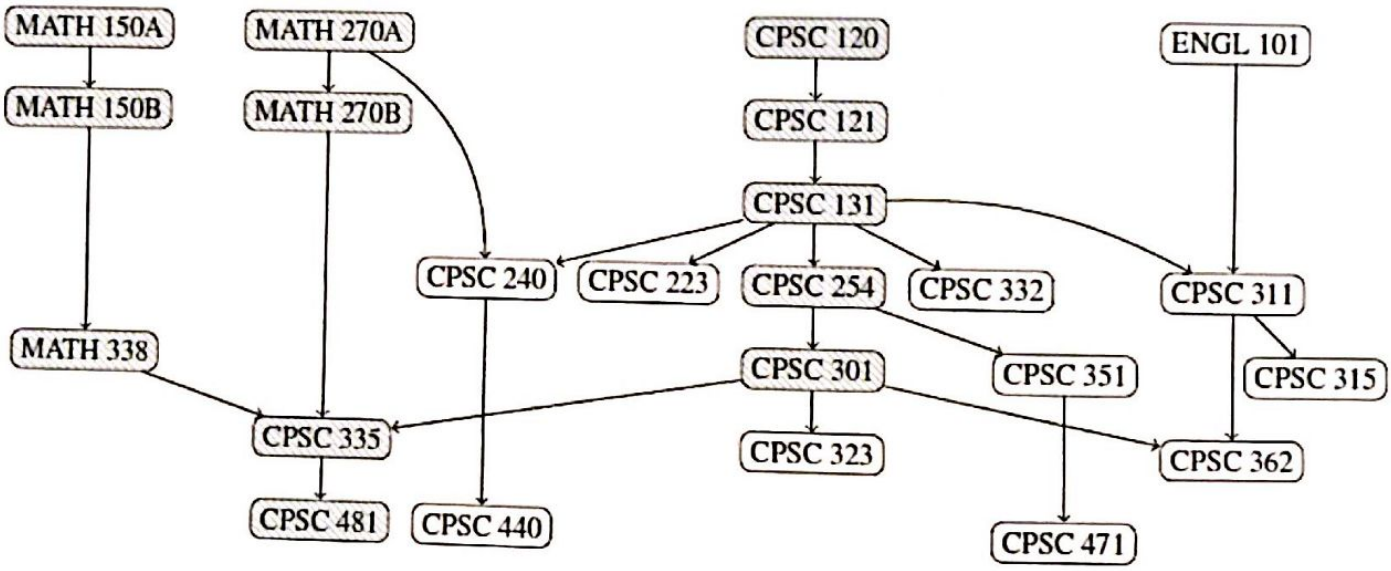
- MICHAEL SHAFAE
- KEVIN WORTMAN

Article Recommendations

DR. WORTMAN

✓ wiki.ecs.fullerton.edu/twinkle/projects
- GrAPH Drawing.org
- LOMBARDI GrAPH Drawings

↑ 26.04.2015



Legend

time-critical course

other course

A → B : A is a prerequisite for B

Article Recommendations

DR. PALMER

- COMPOSITIONAL PATTERN PRODUCING NETWORKS: A NOVEL ABSTRACTION OF DEVELOPMENT, KENNETH O. STANLEY
- EPLEX: EVOLUTIONARY COMPLEXITY RESEARCH GROUP AT UCF
  - EXPLOITING OPEN-ENDEDNESS TO SOLVE PROBLEMS THROUGH THE SEARCH FOR NOVELTY.
- GENERATIVE NEUROEVOLUTION FOR DEEP LEARNING
- THE HYPERCUBE-BASED NEUROEVOLUTION OF AUGMENTING TOPOLOGIES (HYPERNEAT) USERS PAGES
- COMPARISON OF NEAT AND HYPERNEAT ON A STRATEGIC DECISION-MAKING PROBLEM
- 8 SURPRISING FACTS ABOUT REAL DOCKER ADOPTION
- WHY I'M NOT STAKING MY FUTURE ON METEORJS
- METEOR.COM

↑ 10.27.2015

10.28.2015

- THE COMPLEXITY OF MAXIMUM MATROID-GREEDOID INTERSECTION AND WEIGHTED GREEDOID MAXIMIZATION
- INTRODUCTION TO GREEDOIDS
- GREEDOID (WIKIWAND.COM)
- HOW FUNDAMENTAL ARE MATROIDS? GREEDOIDS IN ALGORITHM DESIGN?
- MATCHINGS, MATROIDS & SUBMODULAR FUNCTIONS
- THEORY OF GREEDY ALGORITHMS
- THE COMPLEXITY OF THE MATROID-GREEDOID PARTITION PROBLEM
- THE FORBIDDEN CHARACTERIZATION OF LINE-SEARCH ANTI-MATROIDS OF ROOTED DIGRAPHS
- CORRESPONDANCE BETWEEN 2 ANTIMATROID ALGORITHMIC CHARACTERIZATIONS
- ANTIMATROIDS, BETWEENESS, CONVEXITY
- ON RANK 2 GREEDOIDS

11.09.2015

- GRIDIFIER.IO

11.16.2015

- GRAPH - SIMPLE DATA STRUCTURES AND ALGORITHMS
- LOOM! GRAPH LIBRARY FOR CLOJURE
- GRID LEX. DEVLINT.FI

[redacted]'s Project Proposal: Curriculum Graph Visualizer

2. OBJECTIVES

(1) MODEL CURRICULUM AS A DIRECTED GRAPH
(2) IMPLEMENT CURRICULUM GRAPH AS A WEB APPLICATION

INPUT:
- COURSE NAMES
- PREREQUISITES
- LOCATIONS

OUTPUT:
- GRAPH DRAWING
- MINIMIZE OVERLAPPING EDGES
- ORGANIZE INTO STRATA
- INDICATE COMPLETED COURSES

3. ACTIVITIES

3.1 PHASE I: ALGORITHM

1. ALGORITHM DISCOVERY
2. ALGORITHM TESTING

OUTPUT:
- ALGORITHMS TO IMPLEMENT
- ALGORITHM DESIGN DOCUMENT
  - PSEUDOCODE
  - ALGORITHM SOURCE CODE
  - BRIEF DESCRIPTIONS OF RATIONALE FOR ALGORITHM SELECTION

3.2 PHASE II: DEVELOPMENT

4. ENVIRONMENT

THE FOLLOWING SOFTWARE WILL BE USED IN THE DEVELOPMENT OF THE PROJECT:

1. FRAMEWORKS: Django, BOOTSTRAP
2. Programming LANGUAGES: PYTHON, DOTS
3. WEB LANGUAGES: HTML/HTML5, CSS/CSS3, JAVASCRIPT/JQUERY
4. DATABASE: MongoDB (NOSQL)
5. GRAPH VISUALIZATION: GRAPHVIZ, NETWORKX

5. PROJECT

FINAL RESULTS:

- ALGORITHM DESIGN DOCUMENT
- WEB APPLICATION THAT IS ABLE TO PERFORM THE FOLLOWING:

1. CREATE GRAPH DRAWING
2. ORGANIZE GRAPH INTO STRATA BY SEMESTER
3. ORGANIZE GRAPH INTO TRACKS
4. MINIMIZE OVERLAPPING EDGES
5. INDICATE COMPLETED COURSES

6. SCHEDULE

MARS = SPRING BREAK → CATCH UP!

2016	JAN				FEB				MAR				APR				MAY				SUMMARY	
TASKS:	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	Hours	Pct.
REQUIREMENTS	8	8																			16	5.2%
RESEARCH & DISCOVERY	8	8	8	8	8	8															48	15.6%
DESIGN			8	8	8	8	8	8													48	15.6
INTEGRATE & TEST							8	8	20	20	20	20	20	20	8	8					152	49.4
WRITE USER'S MANUAL															8	8					16	5.2
WRITE FINAL REPORT																	8	8	8		24	7.8
DEMONSTRATE																				4	4	1.3
Hours	16	16	16	16	16	16	16	16	20	20	20	20	20	20	16	16	8	8	8	4	308	100

7. Acknowledgement

- GRADUATE ADVISOR: DR. KEVIN WORTMAN
- GRADUATE REVIEWER: DR. KENT PALMER

□

Drawing Graphs with dot

- By: Gansner et al
- Source: Graphviz.org/Documentation/dotguide.pdf
- DATE: 01.26.2006

ABSTRACT

- Directed graphs drawn in hierarchies
- CMD line prompt
- Web visualization service

1 Basic Graph Drawing

- Graph files or GIF, PNG, SVG, PostScript (→ PDF)
- dot Draws Graphs in 4 phases
  1. Break cycles in input graph by reversing the internal direction of certain cyclic edges
  2. Assigns nodes to discrete ranks or levels
     - In top-to-bottom drawing, ranks determine y coords
  3. Orders nodes within ranks to avoid crossings
  4. Sets x coordinates of nodes to keep edges short
     - Final step routes edge splines
- dot Accepts DOT input
- DOT Describes 3 objects:
  1. Graphs
  2. Nodes
  3. Edges

) dot attributes on pg. 17

Master's Project & Thesis Guidelines

PREREQUISITES

- ✓ REGISTER IN CPSC 597 PROJECT OR CPSC 598 THESIS
- ✓ HAVE CLASSIFIED GRADUATE STANDING
- ✓ FILE AN APPROVED PROPOSAL 1 SEMESTER BEFORE YOU REGISTER FOR 597/598

SUBJECT CATEGORIES

1. SOFTWARE DEVELOPMENT 2. RESEARCH PROJECT	CPSC 597
3. THESIS	CPSC 598

PROPOSING A PROJECT ~~OR THESIS~~

- ✓ IDENTIFY A COMPUTER SCIENCE PROBLEM
- ✓ PERFORM AN ANALYSIS OF THE PROBLEM
- ✓ POSTULATE A SOLUTION
- ✓ INVESTIGATE CURRENT RESEARCH TO SUPPORT YOUR SOLUTION
- SOLVE THE PROBLEM
- TEST THE SOLUTION
- DOCUMENT THE RESULTS
- DEMONSTRATE YOUR PROGRAM OR PRESENT THE RESULTS
- DELIVER THE FINAL PRODUCT

SELECT ADVISOR AND REVIEWER / COMMITTEE

- ✓ FACULTY ADVISOR
- ✓ FACULTY REVIEWER

} MUST BE FULL-TIME FACULTY

PREPARE PROPOSAL

- ✓ OBJECTIVES
  - ✓ GOAL
  - ✓ SIGNIFICANCE
- ✓ ACTIVITIES
  - ✓ HOW OBJECTIVES ACHIEVED
  - ✓ DEVELOPMENT PHASES
  - ✓ TASKS
- ✓ ENVIRONMENT
  - ✓ PROGRAMMING LANGUAGES
  - ✓ COMPUTERS
  - ✓ SOFTWARE TOOLS

REPORTS AND PRODUCTS

- ✓ FINAL REPORT
- ✓ SOURCE CODE
- ✓ USER'S MANUAL

PERFORMING THE WORK

Progress Reports

- Progress reports to your advisor (every 2 weeks preferred)
- Compare actual progress to your schedule
- Discuss problems with advisor
- Adjust schedule if necessary

Draft Documents

- Submit draft documents for advisor to review
- Incorporate comments when preparing the final documents

Demonstration / Presentation

- Demonstrate to advisor
- Demonstrate to reviewer
- Research projects: Give oral presentation

Final Submission

- Submit the final report
- Advisor approval of demonstration
- Reviewer approval of demonstration
- Word or PDF file
- Program source files may come as is
- Program source code in .txt file
- Body paragraph styles:
  - Times New Roman
  - 11 pts
  - Spacing ≥ 13 pts
  - Space after 6 pts
  - No double carriage returns
- For headings:
  - Helvetica
  - 11 point
  - Bold
  - Space before 12pts
  - Space after 3 pts
- Submit 3 final copies
- 8½ x 11 inch paper
- Bound with VeloBind (Kinko's)

Final Evaluation

Your grade on your project is based on your advisor and reviewer's assessment of:

1. The quality of the final report
2. The quality of the product developed.
3. The thoroughness of the individual tasks performed.

Name	Default	Values
bgcolor		background color for drawing, plus initial fill color
center	false	center drawing on page
clusterrank	local	may be global or none
color	black	for clusters, outline color, and fill color if fillcolor not defined
comment		any string (format-dependent)
compound	false	allow edges between clusters
concentrate	false	enables edge concentrators
fillcolor	black	cluster fill color
fontcolor	black	type face color
fontname	Times-Roman	font family
fontpath		list of directories to search for fonts
fontsize	14	point size of label
label		any string
labeljust	centered	"l" and "r" for left- and right-justified cluster labels, respectively
labelloc	top	"t" and "b" for top- and bottom-justified cluster labels, respectively
layers		*id:id:id...*
margin	.5	margin included in page, inches
mclimit	1.0	scale factor for mincross iterations
nodesep	.25	separation between nodes, in inches.
nslimit		if set to *f*, bounds network simplex iterations by *(f)(number of nodes)* when setting x-coordinates
nslimit1		if set to *f*, bounds network simplex iterations by *(f)(number of nodes)* when ranking nodes
ordering		if out out edge order is preserved
orientation	portrait	if rotate is not used and the value is landscape, use landscape orientation
page		unit of pagination, *e.g.* "8.5,11"
pagedir	BL	traversal order of pages
quantum		if quantum ¿ 0.0, node label dimensions will be rounded to integral multiples of quantum
rank		same, min, max, source or sink
rankdir	TB	LR (left to right) or TB (top to bottom)
ranksep	.75	separation between ranks, in inches.
ratio		approximate aspect ratio desired, fill or auto
remincross		if true and there are multiple clusters, re-run crossing minimization
rotate		If 90, set orientation to landscape
samplepoints	8	number of points used to represent ellipses and circles on output (cf. Appendix C
searchsize	30	maximum edges with negative cut values to check when looking for a minimum one during network simplex
size		maximum drawing size, in inches
style		graphics options, e.g. filled for clusters
URL		URL associated with graph (format-dependent)

Table 3: Graph attributes

Name	Default	Values
arrowhead	normal	style of arrowhead at head end
arrowsize	1.0	scaling factor for arrowheads
arrowtail	normal	style of arrowhead at tail end
color	black	edge stroke color
comment		any string (format-dependent)
constraint	true	use edge to affect node ranking
decorate		if set, draws a line connecting labels with their edges
dir	forward	forward, back, both, or none
fontcolor	black	type face color
fontname	Times-Roman	font family
fontsize	14	point size of label
headlabel		label placed near head of edge
headport		n, ne, e, se, s, sw, w, nw
headURL		URL attached to head label if output format is ismap
label		edge label
labelangle	-25.0	angle in degrees which head or tail label is rotated off edge
labeldistance	1.0	scaling factor for distance of head or tail label from node
labelfloat	false	lessen constraints on edge label placement
labelfontcolor	black	type face color for head and tail labels
labelfontname	Times-Roman	font family for head and tail labels
labelfontsize	14	point size for head and tail labels
layer		overlay range for edge
lhead		name of cluster to use as head of edge
ltail		name of cluster to use as tail of edge
minlen	1	minimum rank distance between head and tail
samehead		tag for head node; edge heads with the same tag are merged onto the same port
sametail		tag for tail node; edge tails with the same tag are merged onto the same port
style		graphics options, e.g. bold, dotted, filled; cf. Section 2.3
taillabel		label placed near tail of edge
tailport		n, ne, e, se, s, sw, w, nw
tailURL		URL attached to tail label if output format is ismap
weight	1	integer cost of stretching an edge

Table 2: Edge attributes

Name	Default	Values
color	black	boundary color
comment		any string (format-dependent)
distortion	0.0	distortion factor for shape=polygon
fillcolor	black	shape fill color
fixedsize	false	label text has no affect on node size
fontcolor	black	type face color
fontname	Times-Roman	font family
fontsize	14	point size of label
group		name of node's group
height	.5	height in inches
label	node name	any string
layer		all, id or id:id:id...
orientation	0.0	node rotation angle
peripheries	shape default	number of node boundaries
regular	false	force polygon to be regular
shape	ellipse	node shape; see Section 2.1 and Appendix E
shapefile		external file for custom shape
sides	4	number of sides for shape=polygon
skew	0.0	skew factor for shape=polygon
style		graphics options, e.g. bold, dotted, filled; cf. Section 2.3
URL		URL associated with node (format-dependent)
width	.75	width in inches
z	0.0	z coordinate for VRML output

Table 1: Node attributes

January February March April May June July August September October November December
Monday Tuesday Wednesday Thursday Friday Saturday Sunday
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Graduate Project Meeting Dates

- ADVISOR: DR. KEVIN WORTMAN
- REVIEWER: DR. KENT PALMER
- LOCATION: DR. WORTMAN'S OFFICE CCS
- TIME: 4:00 PM
- DAY: THURSDAYS

PROJECT MEETINGS

✓ #1: FEB 4
✓ 2: FEB 18
· 3: MAR 3
· 4: MAR 17
· 5: APR 7
· 6: APR ~~14~~ 21
· 7: ~~APR~~ MAY ~~28~~ 5
· 8: MAY 12

! · PROJECT DEMONSTRATION: MAY

PURPOSE OF PROGRESS MEETINGS W/ ADVISOR

- COMPARE ACTUAL PROGRESS TO SCHEDULE
- DISCUSS PROBLEMS WITH ADVISOR
- ADJUST SCHEDULE IF NECESSARY

Design: Brainstorming

GRAPH DRAWING

- POSSIBLE OUTPUTS
  - GRAPH FILES; GIF, PNG, SVG; POSTSCRIPT (→ PDF)
- WANT IT TO CREATE WEB OBJECTS THAT CAN BE MANIPULATED BY JAVASCRIPT

COURSE INPUT (DATA)

- INPUT FIELDS (WEB FORM)
- FILE IMPORT (CSV, TXT, ETC.)
- DATABASE PULL ?
- JSON TRANSLATED FROM INPUT
- MARK COMPLETED COURSES

USER INTERFACE

- DATA INPUT
- GRAPH LAYOUT

ALGORITHM

- DATA ORGANIZATION
- LABELING
- ALPHABETIZATION?
- RANK & FILE ASSIGNMENT

GraphViz for Discrete Math Students

- GENERATE UNDERGRAD CURRICULUM GRAPH INTO GRAPHVIZ/DOT
- PRINT OUTPUT + SOURCE CODE

NOTES

- ORGANIZING BY RANKS (columns) AND ~~ROWS~~ FILES (ROWS) MAY BE MORE DIFFICULT TO WRITE INTO AN ALGORITHM.
- VERTEX VARIABLE NAMING CONVENTION
  - BY LETTER → 24 for single char
  - BY COURSE NAMES / ID → MORE DESCRIPTIVE
  - BY NUMBER → UP TO 99 POSSIBILITIES IF DOUBLE DIGIT
- PRINT OUT "IMPORTANT ATTRIBUTES" LIST
- rank=same DOES NOT SEEM TO WORK WITH SUBGRAPHS
  - clusterrank MAY SOLVE THAT PROBLEM!
- DOES GRAPHVIZ HAVE EDGE CROSSING DETECTION?
- EVEN AFTER INPUTTING ALL THE COURSES & PREREQ'S, THERE WAS STILL ONE CROSSED EDGE...

2/1/2016 4989392942792704 (1144×635)