DSA Final Project Proposal

Dieter Brehm & Elias Gabriel

5 April 2020

letshwit!

a homework scheduler

Goals

- **Elias**:
  Develop a better intuition for the structural, runtime, and algorithmic properties of generic heaps Learn more about how to build reliable and stable APIs
- **Dieter**
  Learn about connecting algorithms to built-out solutions that help people Explore user interaction in algorithm development

Ideas

We are planning to implement a homework scheduling system which assists students in optimizing when to study. It can be difficult to balance NINJA hours, class due dates, and lengthy assignments, but using a heuristic-based priority queue in the context of time, we will try and give suggestions for how to improve the studying experience. Possible parameters to our weighting system could includes due dates for specific homework, time entry from the users, a user's current grade on canvas, when class is happening, ninja-hours optimization, and desired sleeping hours.

We see parallels to the knapsack and other pathfinding problems, and also in concepts used in CPU scheduling. We're both taking softsys currently, and see connections to computer architecture talked about there.

References

An overview of the A* algorithm: https://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html. This will help build our foundational knowledge on heuristic-based path finding.

A heuristic-based approach to optimal path finding:https://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html. We need to know about heuristics in-depth in order to figure out how we may be able to adapt it to our own algorithm.

A comprehensive overview of heuristic functions:https://medium.com/@rinu.gour123/heuristic-search-in-artificial-intelligence-python-3087ecfece4d Algorithms like A* do not guarantee that they'll find the globally best solution, but do guarantee that they'll find the best solution for the given heuristic functions. Therefore, resources like these will be critical in learning more about how to select better and more optimal functions.