

Movie-Chain-Runner Problem

Shashank Singh Jimmy Zong

June 6, 2013

Outline

- 1 Team Members
- 2 Problem Introduction
- 3 Benefits
- 4 Approach
 - Algorithms
 - Project Timeline
- 5 Evaluation
- 6 Qualifications
- 7 Summary

Team Members

- Sung Uk Ryu
- Eugene Scanlon
- Shashank Singh
- Jimmy Zong

The Problem

The Problem

Find the “longest” list of overlapping titles in a list of movie titles.

The Problem

The Problem

Find the “longest” list of overlapping titles in a list of movie titles.

For Example: In the list

- Day of the Dead
- Live and Let Die
- Dead Poets' Society
- Die Another Day
- The Last Samurai

the “longest” chain is

“Live and Let Die Another Day of the Dead Poets' Society.”

The Problem

- Equivalent to finding a Longest Path in a directed graph

The Problem

- Equivalent to finding a Longest Path in a directed graph
- The Longest Path Problem is NP-hard, meaning that there is no efficient algorithm for solving it on a large graph

Previous Attempts

- Summer 2010 – 255 titles
- Fall 2010 – 271 titles (845 words)
- Summer 2011 – 311 titles (997 words)
- Fall 2011 – 323 titles (1030 words)
- Spring 2012 – 327 titles (1055 words)

Benefits

The members of our group will gain experience

- programming in Python (and maybe C or MATLAB)
- working as a group toward a common goal
- handling and processing a large data set
- implementing graph algorithms
- designing and implementing approximation algorithms for an NP-hard problem

Approach

- Algorithms
- Project Timeline (Gantt Chart)

Algorithms

① Brute Force

- Tried running on 16 GHC machines for 15 hours
- Constructed chain of 247 titles
- Progress slowed exponentially

Algorithms

1 Brute Force

- Tried running on 16 GHC machines for 15 hours
- Constructed chain of 247 titles
- Progress slowed exponentially

2 Acyclic Subgraphs

- A poly-time topo-sort algorithm is known for acyclic graphs
- Try to find acyclic subgraphs
- Too many cycles – took too long to generate subgraphs

Algorithms

1 Brute Force

- Tried running on 16 GHC machines for 15 hours
- Constructed chain of 247 titles
- Progress slowed exponentially

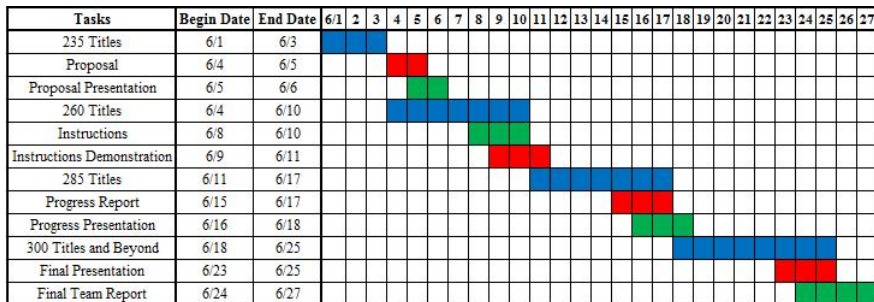
2 Acyclic Subgraphs

- A poly-time topo-sort algorithm is known for acyclic graphs
- Try to find acyclic subgraphs
- Too many cycles – took too long to generate subgraphs

3 Working backward

- Stuck at 247 titles using brute force
- Reverse graph edges and add to the beginning of the chain
- Work in progress – currently managed 274 titles

Gantt Chart



Key:	
Eugene/Sung	█
Shashank/Yiming	█
Everyone	█

Evaluation

- 1 Length of the longest chain we find

Evaluation

- 1 Length of the longest chain we find
- 2 Compare performance of a different algorithms
 - ideally, decide on a “best” algorithm for the problem

Evaluation

- 1 Length of the longest chain we find
- 2 Compare performance of a different algorithms
 - ideally, decide on a “best” algorithm for the problem
- 3 Predict runtime for entire computation by solving tractable subproblems and extrapolating

Qualifications

All members of our team have programming experience in Python.

- Sung Uk Ryu
 - Junior Computer Science and Finance Double Major
 - Experience with scheduled, task-oriented projects
- Eugene Scanlon
 - Junior CS Major with minor in Music Technology
 - Additional programming experience in C0, C, and Nyquist

Qualifications

- Shashank Singh
 - Senior CS/Math Dual Degree
 - Have TA'd 15-211 and 15-251
 - Experience analyzing large data sets with C and MATLAB
- Jimmy Zong
 - Sophomore CS major
 - Experience with BASH scripting and C programming
 - Experience running distributed computations on a UNIX server

Summary

- 1 Team Members
- 2 Problem Introduction
- 3 Benefits
- 4 Approach
 - Algorithms
 - Project Timeline
- 5 Evaluation
- 6 Qualifications
- 7 Summary

Sources

- Gantt Chart created using software from the Gantt Project
 - <http://www.ganttproject.biz/> (accessed June 4, 2013)
- Git repository hosted on GitHub
 - <https://github.com/>