Contents

Facilitator Information

Learning Objectives (for content & process)

After completing this activity, learners should be able to:

- Explain the pros & cons of various simple search strategies.
- Explain common tradeoffs between the complexity and performance of algorithms.
- Assess the performance of simple algorithms as a function of their input size N.

This activity should help learners develop teamwork and critical thinking skills.

Prerequisites (for content & process)

Before starting this activity, learners should have:

- Previous experience with POGIL (useful but not necessary).
- Programming experience is not required for the activity, but is for some applications.

Preparation

1. Optional: Provide the worksheet on the board, a poster, or in presentation software, so teams can see each others' work easily.

Activity Notes

- If time is limited, skip: III.1, III.4, IV.
- The facilitator should spend a minute or two introducing the activity.
- While student teams work, the facilitator should circulate among the teams to monitor progress and help with problems, although the facilitator should avoid providing or confirming answers to any of the key questions.
- I.3. Report out: describe a strategy. Summarize in table for class.
- II.4. Report out: rankings. Summarize in table for class.
- III.4. Report out: worst & average case results. Summarize in table for class.

Things to Do

- Review & test
- Part IV is too complicated, at least for typical CS1 students.
- Add more guidance on mean & max counts
- Add sections on: implications for search: effort required to sort, tradeoffs.
- Add applications, resources, rubrics
- Add summary section.
- Add supporting resources.

Activity History

2010-06	drafted by Clif Kussmaul clif@kussmaul.org
2010-07..08	split into 2 activities (search and O() notation), revised
2011-01	piloted, revised
2012-12	revised

start time:

Introduction

In computing, we often must **search** in a set for a particular item. As computer scientists, we are particularly interested in searching very large sets, with thousands or millions of values. For example, the Harvard University Library has roughly 16,000,000 volumes, and the US Library of Congress has roughly 22 million cataloged books, and over 100,000,000 total items. In this activity, we use a simple game to explore some basic searching algorithms. This will also help us explore more general concepts in algorithm design and analysis, so studying searching is useful even though very few of us may need to implement searching algorithms, since efficient techniques are part of most software libraries (APIs).

Before you start, complete the form below to assign a role to each member.
If you have 3 people, combine Manager & Reflector.

Team		Date	
Team Roles		**Team Member**	
Recorder: records all answers & questions, and provides copies to team & facilitator.			
Speaker: talks to facilitator and other teams.			
Manager: keeps track of time and makes sure everyone contributes appropriately.			
Reflector: considers how the team could work and learn more effectively.			
Other:			

Reminders:

1. *Note the time whenever your team starts a new section or question.*
2. *Write legibly & neatly so that everyone can read & understand your responses.*

Hi-Lo Game

Hi-Lo is a number guessing game with simple rules.

a. There are two players – A and B.
b. Player A thinks of a number from 1 to 100.
c. Player B guesses a number.
d. Player A responds with “too high”, “too low”, or “you win”.
e. Players B and A continue to guess & respond until B wins (or gives up).

start time:

I. (10 min) Player Strategies

1. (3 min) Play the game a few times to ensure that everyone understands the rules.

2. (2 min) List up to 3 ways to clarify the rules.

3. (3 min) Describe 4-5 different strategies that Player B could use to guess numbers.
Try to have a mixture of simple and clever strategies.
Name each strategy and list it in the first column of the worksheet.
Before you continue, review progress with the facilitator.

start time:

II. (10 min) Comparing Strategies

1. (2 min) Evaluate each strategy with regard to how **quickly** it will find the right answer,
by rank ordering from 1 (least guesses) to 5 (most guesses).
Add the rankings to the worksheet in a column labeled **Quick**.

2. (2 min) Evaluate each strategy with regard to how **easy** it is to describe or specify,
by rank ordering from 1 (easiest) to 5 (hardest).
(Suppose you had to explain each strategy to a first-grader so that she could play the game.)
Add the ranking to the worksheet in a column labeled **Easy**.

3. (1 min) For each strategy, multiply the quick rank by the easy rank,
and add the product to the worksheet in a column labeled **Product**.

4. (3 min) In complete sentences, describe the relationships between the two sets of rankings.
Before you continue, review progress with the facilitator.

start time:

III. (10 min) Worst & Average Case Performance

1. (2 min) Discuss and list the pros & cons of measuring program speed with a stopwatch.

2. (3 min) For each strategy, determine the **worst case** (maximum) number of guesses required to win.
Add the numbers to the worksheet in a column labeled **Worst**.

3. (3 min) For each strategy, determine the **average case** (typical) number of guesses required to win.
Add the numbers to the worksheet in a column labeled **Average**.
Note that the **minimum** number of guesses is always 1 – it's nice to be lucky.

4. (2 min) List 3 reasons why it would be useful to have more precise, quantitative ways
to measure and discuss the speed of an algorithm.
Before you continue, review progress with the facilitator.

start time:

IV. (10 min) Effect of Input Size

1. (3 min) Assume that Player A chooses a number from 1 to 1000.
For each strategy, what are the worst case & average case number of guesses?
Add the numbers to the worksheet in columns labeled “1K Worst” and “1K Average”.

2. (4 min) **Optional**: Assume that Player A chooses a number from 1 to N.
(For example, N=100, N=1000, N=1,000,000)
For each strategy, what are the worst case & average case number of guesses in terms of N?
Add the expressions to the worksheet in columns labeled “N Worst” and “N Average”.
(Hint: you’ve already done N=100 and N=1000; consider other values before generalizing to N.)

3. (3 min) Describe the pros & cons of analyzing performance in terms of input size N.

Applications

1. Team Meeting Minutes (TMM) – see common description (separate doc) and rubric (below).

2. Personal Reflection Memo (PRM) – see common description (separate doc).

3. Write a program (or set of programs) to play the game using each strategy.
The program should count the guesses and print the total when the game ends.
Record how much time it takes to code each strategy. Is this consistent with your “Easy” rating?
Play the game 5 times with each strategy. Is the guess count consistent with your “Quick” rating?

Team Meeting Minutes (TMM): Rubric

SPECIFIC CRITERIA	RATING	COMMENTS
I: Comprehensively synthesizes guessing strategies.	/ 4	
II: Comprehensively synthesizes rankings & tradeoffs.	/ 4	
III: Comprehensively synthesizes worst & average case performance.	/ 3	
IV: Comprehensively synthesizes N-analysis.	/ 3	
COMMON CRITERIA	**RATING**	**COMMENTS**
Begins with a summary of the activity; ends with a summary of questions, and a list of action items, if needed.	/ 3	
Mechanics & format: Work is neat, well organized, & clearly provides all required information, including team member roles & timing data.	/ 3	
TOTAL	/ 20	

Searching

2013-01-03

Facilitator – Answer Key

#	Player Strategy	II.1. Quick	II.2. Easy	II.3. Prod	III.2. Worst	III.3. Avg	IV.1. iK Worst	IV.1. 1K Aver	IV.3. N Worst	IV.3. N Aver
1	Guess numbers at random, ignoring “hi-lo” feedback, without memory (repeat guesses possible)	5	2	10	1K	50	10K	500	10N	N/2
2	Guess numbers at random, ignoring “hi-lo” feedback, but don’t repeat guesses	4	3	12	100	50	1000	500	N	N/2
3	Count up from 1 (or down from 100)	3	1	3	100	50	1000	500	N	N/2
4	Count up by 10s, then down by 1s (many variations)	2	4	8	20	10	110 (30)	55 (15)		
5	Guess 50, then 25 or 75, then 12 or 38 or 62 or 88 – keep track of min & max possible values, divide range in half	1	5	5	7	6	11	10	$\log_2 N$	$\log_2 N$

Searching: Worksheet

#	Strategy Name	Quick	Easy	Prod	Worst	Average	1K Worst	1K Average	N Worst	N Average
1										
2										
3										
4										
5										
6										