APP2: Hold'em for n00bs

Algorithmic Problem Solving

Florent Bouchez Tichadou — Manuel Selva
florent.bouchez-tichadou@imag.fr,manuel.selva@inria.fr

September 28th, 2018

Source: http://novicepoker.wordpress.com

Learning objectives:

- Understand the limitations of greedy algorithms.
- Use dynamic programming to find an optimal solution to a problem.
- Identify the cases where dynamic programming can be used.

I. APP2: Hold'em for n00bs

I.1. APP Objectives

After this APP2, you will be capable of:

- Knowing the principles of the following various types of problem-solving techniques: greedy algorithms, complete solution space exploration and dynamic programming;
- Evaluating the optimality and complexity of these methods on problems;
- Identifying the cases where dynamic programming is applicable and use it to get an optimal solution.

I.2. Session Organization

This APP will spread on six sessions. Time between session is for personal work ("trap", for *TRAvail Personnel* in french).

Session 1. Opening (40 min.) Discovery of the problem by personal reading then group discussions.

Trap 1. ⇒ Read provided material on greedy algorithms. Write (on paper) a greedy algorithm to find a simple (albeit non-optimal) solution to the problem. Try it on some examples and select some where your algorithm is working and some where it is completely false.

Session 2. (1h30) Sharing work in group about the greedy algorithm resolution of the problem and the prepared examples. Decide collectively on the best algorithm to do it.

Trap 2. ⇒ Write (on paper) an exploration algorithm to find the best solution of the problem. Evaluate the complexity of such an algorithm with increasingly more cards. What happens with a 52-card deck?

Session 3. (1h30) Each student presents his/her algorithm; Election of the best one.

Trap 3. ⇒ Reading of the rest of the notes: dynamic programming.
Important: Prepare at least two–three questions to ask the tutor on the reading material for next session.

Session 4. (1h30) Restructuring session : Tutors will help answering questions of the class. In groups, you will then define the principles to apply the dynamic programming method to the problem.

Trap 4. ⇒ Write an algorithm based on dynamic programming.

Session 5. (1h30) Share your work with the group and write in group an algorithm that solves the problem. Evaluate its complexity.

Trap 5. ⇒ Continue working on the dynamic programming algorithm or work on the optional part.

Session 6. Closing (1h30) Sum up all step in a document. This deliverable must be given to the tutors by email at the end of the session.

Important: Between each and every session, personal work is expected.

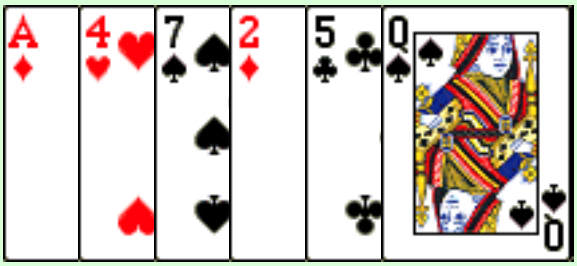
I.3. Situation—problem

You are playing cards with your 6-year old little sister or brother or nephew or your neighbour's son, or a dog or whatever, just find someone ok?
Your ultimate goal is to strip her of all of her money at poker, sadly, she's far too young for that and her income is not yet sufficient to support your grand student's life, so you resign and start playing a much more simple game instead:

> This is a 2-player game. There is a series of n cards lying on the table face up arranged in a line. Players take turns to take cards one at a time, but only the rightmost or the leftmost card can be picked of the line.
>
> When the last card is taken by a player the game is finished. We sum the points of all cards taken by each player (aces are worth 14!). The player with the highest score wins the game.

Given the fact that your sister (or dog) is really young it should be quite easy for you to beat her. However, she is not as dumb as you thought, and she's playing using a very simple scheme that seems to work quite well. She always choses the card with the highest value she can get. Even with this elementary strategy, she beats you in about half the games.

You have to elaborate a smarter strategy to increase your winning ratio at this game, even if your sister always chooses who goes first.



I.4. Resources

This APP will cover basic knowledge on greedy algorithms, solution space exploration and dynamic programming. It is required that you read the notes given at the address below. Moreover, if you feel you need more information, the book reference is extremely complete and give great details on this subject.

- https://im2ag-moodle.e.ujf-grenoble.fr/course/view.php?id=223 ⇒ greedy-algo.pdf
- https://im2ag-moodle.e.ujf-grenoble.fr/course/view.php?id=223 ⇒ dynamic-programming.pdf
- "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein (3rd edition);
  - Chapters 16 (Greedy algorithms);
  - Chapter 15 (Dynamic Programming);

I.5. Roles

For this APP, inscribe the name of the volunteers for each of those roles (do not take a role you have already been!):

Chair _____
Scribe _____
Time keeper _____
Recorder _____

II. Session 1: Discovery and Analysis of the Problem (40 min.)

Timing	*What to do during the session*
1 5 min.	Organize your group: attribute roles (chair, scribe, recorder) for the project. People must volunteer but not take a role already taken in the previous APP.
2 15 min.	**Warning:** individual phase (no discussion!) First approach of the problem: read the problem with care. Try to find by yourself elements of solutions to the problem.
3 15 min.	Share in the group your comprehension of the problem and your first ideas. Write on the board (or a common sheet of paper) all your thinking, while keeping a special place where to put element that require precisions by the tutor. Try to organize your thinking around the following points: • Try some examples game with the playing cards you brought. • Focus on modelling the problem.
4 5 min.	Make sure to understand what is expected for the next session.

Total: 40 min.

III. TRAP 1: Greedy algorithm solution (expected work: 1h)

We remind you that personal work for next session is **required** and *not an option*!

- Read with attention the first document of the notes on greedy algorithm (see Section I.4). If you feel you need it, also consult the chapters of the book mentioned.
- Elaborate individually an algorithm that simulates the game when you play using the same naive (greedy) solution that your sister uses. Test on some examples and **find working examples and non-working ones**. Be careful to describe clearly each step of your reasoning so it can be easily explained (and understood) by the other members of your group. If this part is easy for you, start thinking about the next TRAP.

Warning: Important: it is crucial that everyone in the group does its share of work, as otherwise no work can be done during the next session. You are responsible for **your own learning**, but the learning of the other members of your group **depends also on you**!

IV. Session 2: Work Session (1h15)

Timing	*What to do during the session*
1 20 min.	Share the algorithms and examples you have come up with in your group. **Important:** Everyone must present something! It is the chair's duty to make sure of that.
2 50 min.	From your sharing, elaborate collectively a relevent greedy algorithm, and elaborate on its limitations and advantages.
3 5 min.	Make sure everyone understands the work for next session.

Total: 75 min.

V. TRAP 2: Complete solution space exploration (expected work: 1h)

For next session, every member of the group needs to write an algorithm (on paper) exploring the complete solution space to find the best solution. Study the complexity of this solution and its practicality for a high number of cards (e.g., a full 52-card deck).

VI. Session 3: Work session (1h15)

Timing	*What to do during the session*
1 20 min.	Share the algorithms proposed by group members. Do not forget to exchange on the complexity of your solutions. **Important:** Everyone must present something! It is the chair's duty to make sure of that.
2 50 min.	From your sharing, elaborate collectively an optimal algorithm to the problem and evaluate the complexity of this algorithm and its applicability with a high number of cards.
3 5 min.	Take the time to understand what is required for next session.

Total: 75 min.

VII. TRAP 3: Dynamic programming (expected work: 1h30)

We remind you that personal work for next session is **required** and *not an option*!

- Read with attention the document on dynamic programming. If you feel you need it, also consult the chapters of the book mentioned. To help you during your reading, keep notes on how this relates to the current problem of the APP.

- Dynamic programming is a difficult concept to master. In all likelihood, you will not fully grasp the all of it with just one reading.
  Important: Read the notes until you can point specifically where you have difficulty understanding.

- The next session will be a restructuring lecture based on your questions.
  Important: Write down at least 2-3 precise questions, and **send them by email to both tutors** before Sunday 23:59. We will answer them during the next class (on Monday).

VIII. Session 4: Restructuring Lecture (1h30 max.)

Timing	*What to do during the session*
1 90 min.	Discussion with the tutors based on the questions received by email. **Important:** Warning: No question, no lecture!

Total: 90 min.

IX. TRAP 4: Define a solution based on dynamic programming (expected work: 2h)

For next session, every member of the group needs to write (on paper) an algorithm based on dynamic programming to find the best score you can achieve on a game with your sister.

X. Session 5 and 6: Work session (2 * 1h30)

By this time, you should have learned how to manage your time and efforts. These last two sessions will be devoted to having a working dynamic programming algorithm that solves the problem and finishing the report.

If the algorithm is not clear for everyone in the group, it is crucial to take the time to explain to each other what it does and how it works.

If you have more time to go further, we advise you to continue working of the following points (in this order):

1. Write a formal proof that dynamic programming finds an optimal solution to the problem (use the lecture notes);

2. Write an algorithm that solves the problem when both players are playing optimally.

Important: The report must be handed to your tutor before the end of the last session!

Don't forget to do as always an individual assessment of your learning by filling the chart on page 7. Those points will be testing during the evaluation at the beginning of the next session.

XI. Closing Session

Timing	*What to do during the session*
1 20 min.	Individual evaluation to test learning on APP2.
2 20 min.	Group analysis. • **Individually:** fill the circept and answer the questions on team work (see next page). • Compare your results with the others members and draw conclusions. • Share your analysis with the tutor.
3 50 min.	*Discovery of the next APP...*

Total: 90 min.

XII. Your Learnings

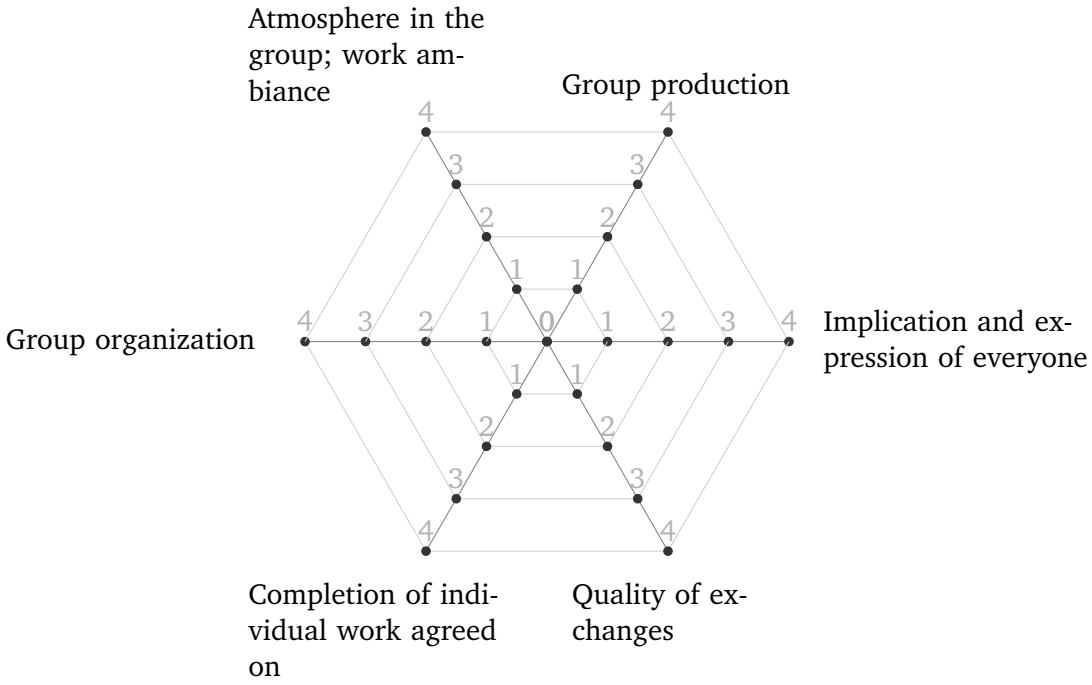
For each objective of this APP, evaluate by yourself your level of learning.

After this APP, you are capable of:	*Yes*	*No* (need to work on it)
1. Defining what a greedy algorithm is and writing one;	_____	_____
2. Know the limitations of greedy algorithms and find counter examples to their optimality;	_____	_____
3. Write an optimal full space exploration algorithm and analyse its complexity;	_____	_____
4. Identify the cases where dynamic programming can apply;	_____	_____
5. Write a dynamic programming algorithm and analyse its complexity.	_____	_____

XIII. Auto-Evaluation of the Team Work — Circept

XIII.1. Assessment of group work

- **Individually:** fill the circept below and answer the questions on team work (10 min.)
- Compare your results with the others members and draw conclusions.
- Share your analysis with the tutor.



XIII.2. Give two positive points about your group work:

-
-

XIII.3. Give two negative points about your group work:

-
-

XIII.4. What commitment can you take to improve your group work?

-