

**Universiti Teknologi MARA**

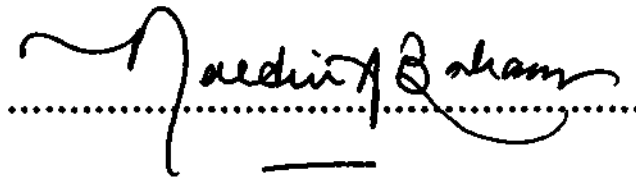
**Comparison between Alpha Beta Pruning  
Algorithm and Greedy Algorithm in Designing  
Winning Strategies in the Game of Checkers**

**CHRISTY JR TAN**

**Final year project report submitted in part fulfilment  
of Bachelor in Computer Science (Hons.)  
Faculty of Information Technology and  
Quantitative Sciences**

04 July 2007

**SUPERVISOR'S APPROVAL**

A handwritten signature in black ink, reading "Nordin Bin Abu Bakar", is written over a horizontal dotted line. Below the dotted line, there is a short horizontal solid line.

**SUPERVISOR'S NAME : PM DR. NORDIN BIN ABU BAKAR**

**DATE : 6<sup>TH</sup> JULY 2007**

## **DECLARATION**

I certify that this final year project report and the research to which it refers are the product of my own work and any ideas or quotation from the work of other people, published or otherwise are fully acknowledged in accordance with the standard referring practices of the discipline. I, hereby declare that I am responsible for the content of this report as it had been submitted as part of partial fulfillment of Bachelor in Computer Science program.

JULY 4<sup>th</sup>, 2007

CHRISTY JR TAN

2005614243

## **Abstract**

Alpha Beta Pruning is an algorithm to prune unnecessary branches. The idea of not exploring the branches if we know that it is worthless makes it a powerful algorithm ever invented. The algorithm is useful in some of the game search available today. Meanwhile, the ability of greedy algorithm to solve problem by choosing any alternatives that leads to the optimal solution without having care of what will happen after that as long as it achieves its goal makes it suitable in any game it applies. Some of the games are checkers. Checkers uses game search to find the solution to win the game. There are many available locations that the pieces can move in the board. It makes it difficult for the player to determine the next move that it should go to win the game. This game will be used to compare both algorithm based on time and the number of moves it takes to win the game. By applying both algorithms, we can determine which algorithm is the most powerful algorithm to generate the winning strategies in the game of checkers. The result will also show how the algorithms generate the winning solution.

## List of Figures

Figure 2 <i>Minimax Search Tree</i>	8
Figure 3 <i>Example of alpha-beta pruning.</i>	9
Figure 4 <i>Heuristic measuring conflict applied to states of tic-tac-toe from Luger(2005)</i>	10
Figure 5 <i>Two-ply minimax applied to X's move near the end of the game from Nilsson (1971)</i>	10
Figure 6 <i>The 3 Last One Loses near the end of the game.</i>	11
Figure 7 <i>Roadmap in search problem</i>	13
Figure 8 <i>Minimum Spanning Tree (Skiena,1997)</i>	16
Figure 9 <i>Kruskal Illustration</i>	17
Figure 10 <i>The Methodology Phases</i>	23
Figure 11 <i>The Feasibility Study Model</i>	25
Figure 12 <i>The Game Model</i>	26
Figure 13 <i>Project Design</i>	27
Figure 14 <i>Data Model</i>	28
Figure 15 <i>Project Framework</i>	29
Figure 16 <i>Flow Chart</i>	31
Figure 17 <i>Data Collection Process</i>	32
Figure 18 <i>Components of the Interface</i>	36
Figure 19 <i>A message box display the winner of the game</i>	37
Figure 20 <i>Alpha Beta Pruning analysis based on Time</i>	43
Figure 21 <i>Average number of time winning in Alpha Beta Pruning</i>	44
Figure 22 <i>Alpha Beta Pruning analysis based on number of Moves</i>	45
Figure 23 <i>Average number of moves in Alpha Beta Pruning</i>	46
Figure 24 <i>Average number of games won in Alpha Beta Pruning</i>	47