**Part 1: Constraint Satisfaction Problems**

**Graduation** **Dinner**

1. CSP Formulation:
   1. Variables
      1. J – Jasmine
      2. S – Jason
      3. T – Trudy
      4. R – Randy
      5. M – Misty
   2. Domains
      1. 1 – 5 where 1 is closest to kitchen
   3. Constraints
      1. Alldiff
      2. notAdjacent(J, S)
      3. S < J
      4. notAdjacent(M, S)
      5. notAdjacent(M, T)
      6. notAdjacent(M, R)
      7. |M – T| > 2
2. Arc Consistency:
   1. J : 3, 4, 5
   2. S : 1, 2, 3
   3. T : 1, 2, 4, 5
   4. R : 1, 2, 3, 4, 5
   5. M : 1, 2, 4, 5
3. MRV: Both J and S have 3 legal variables, so either J or S would be assigned first.
4. R = 3 Arc Consistency:
   1. J : 4, 5
   2. S : 1, 2
   3. T : 1, 2, 4, 5
   4. R : 3
   5. M : 1, 5
5. R = 3 Solutions:
   1. T S R J M
   2. S T R J M

**Hide & Seek**

1. Simple Backtracking
   1. CSP Formulation
      1. Variables : v1, v2, … , vN
      2. Domains: {(X, Y) | X,Y ϵ [1, N]}
      3. Constraints
         1. Alldiff
         2. If ( vI=(X, yI) && vJ=(X, yJ) ) then tree=(X, yK) | yk ϵ (yI, yJ)
         3. If ( vI=(xI, Y) && vJ=(xJ, Y) ) then tree=(xK, Y) | xK ϵ (xI, xJ)
         4. If ( vI=(X, Y) && vJ=(X±c, Y±c) ) then tree=(X±k, Y±k) | k ϵ (0, c)
   2. Backtracking approach explained: The file hideNseek.py features a program to solve the hide and seek constraint satisfaction problem. This program attempts to place each friend incrementally into some location. If there are no legal locations left, the algorithm backtracks by re-positioning the last placed friend.
   3. Number of attempted variable assignments: The input files can be found in the folder and are named in the form <Trees>in<Friends>.txt. You can run the algorithm using these input files.

For example: python hideNseek.py –s 32in14.txt –h localManhattan –p

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Null Heuristic | | Local Manhattan | | Global Manhattan | |
| Friends | Trees | Nodes | Backtracks | Nodes | Backtracks | Nodes | Backtracks |
| 4 | 5 | 29 | 0 | 9 | 0 | 7 | 0 |
| 5 | 10 | 61 | 0 | 18 | 0 | 18 | 0 |
| 6 | 14 | 83 | 0 | 30 | 0 | 26 | 0 |
| 7 | 17 | 116 | 0 | 45 | 0 | 66 | 0 |
| 8 | 15 | 190 | 0 | 83 | 0 | 49 | 0 |
| 9 | 18 | 274 | 0 | 79 | 0 | 109 | 0 |
| 10 | 22 | 341 | 0 | 99 | 0 | 126 | 0 |
| 11 | 25 | 448 | 0 | 95 | 0 | 143 | 0 |
| 12 | 26 | 574 | 0 | 184 | 0 | 355 | 0 |
| 13 | 29 | 695 | 0 | 127 | 0 | 336 | 0 |
| 14 | 32 | 814 | 0 | 175 | 0 | 319 | 0 |

Because the program never backtracks for any input, a plot of backtracks versus friends is trivial and left absent.

1. Heuristics
   1. The number of nodes expanded by the program using the local Manhattan and global Manhattan heuristics is shown in the table above. The program did not need to backtrack when using any heuristic, but the number of nodes expanded when using either heuristics was much less than using no heuristic.

**Part 2: Minimax:**

**Candy Game**

The python file candyGame.py implements a game environment as well as agents to play the candy game. Any pair of human, minimax, or alphabeta players may play against each other. Search depths for the computer agents may be specified, but they default to 3 for the minimax agents (minimax4 per turn expanded 1.5 million nodes in 6 minutes) and 4 for the alphabeta agents (alphabeta5 per turn expanded 1.5 million nodes in 7 minutes).

Both minimax and alphabeta agents use the same evaluation function for non-terminal nodes. The evaluation function is designed to return a linear probability of winning based on the game state. If the game is over, a 1 is returned for wining, and a 0 is returned for losing. If neither player has any points, 0.5 is returned. Otherwise, the fraction of total points of both players belonging to the evaluating player is returned.

A table of nodes expanded and final scores is given below for each of the four match-ups on each game board. For brevity’s sake, the sequence of moves, final board state, and total time each player took is available at the end of this document. The average number of nodes and the time taken is directly related to the total number of nodes expanded.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Board** | **Metric** | **Minimax** | **Minimax** | **AlphaBeta** | **AlphaBeta** | **AlphaBeta** | **Minimax** | **Minimax** | **AlphaBeta** |
| Almond Joy | Points | 15 | **21** | 17 | **19** | 17 | **19** | 14 | **22** |
| Nodes | 217740 | 194736 | 524874 | 428028 | 524874 | 194736 | 217740 | 374922 |
| Ayds | Points | **1197** | 603 | 803 | **997** | **999** | 801 | 802 | **998** |
| Nodes | 217740 | 194736 | 622536 | 687009 | 529986 | 194736 | 217740 | 660043 |
| Bit-O-Honey | Points | **210** | 168 | **191** | 187 | **212** | 166 | 106 | **272** |
| Nodes | 217740 | 194736 | 4610363 | 4097692 | 4865132 | 194736 | 217740 | 3894960 |
| Mounds | Points | 33 | **39** | 33 | **39** | 34 | **38** | 29 | **43** |
| Nodes | 217740 | 194736 | 1187759 | 1093397 | 1210471 | 194736 | 217740 | 1188439 |
| Reeses Pieces | Points | 633 | **1020** | **933** | 720 | **983** | 670 | 826 | **827** |
| Nodes | 217740 | 194736 | 1063581 | 1363347 | 1083505 | 194736 | 217740 | 1544871 |
| Oases | Points | 10 | **30** | **26** | 14 | **30** | 10 | 18 | **22** |
| Nodes | 217740 | 194736 | 577443 | 702201 | 544068 | 194736 | 217740 | 753673 |
| Long | Points | **43** | 26 | **37** | 32 | **35** | 34 | 32 | **37** |
| Nodes | 299080 | 269800 | 1502260 | 1419671 | 1374885 | 269800 | 299080 | 1458470 |
| Small | Points | **9** | 3 | **9** | 3 | **9** | 3 | **9** | 3 |
| Nodes | 945 | 600 | 2471 | 1379 | 2471 | 600 | 945 | 1379 |

**Bonus Points**

An interface for playing against either type of computer agent is easily accessed by specifying a human player. For example, “$python candyGame.py –p1 human –p2 minimax” will initialize a game on the ReesesPieces game board between a human and the minimax agent. Using the standard evaluation function presented above, it is very difficult to beat the AI agents with search depths 3 or greater. An agent using a search depth of 2 is not easily beaten on complicated maps.

Three custom boards are included in the game\_boards folder, and the results of the AI matchups on these boards are presented in the table above. Oases.txt is the most interesting because if both agents use search depth of 3, then the second player wins. If both agents use a search depth of 4, then player one wins.

I also implemented Quiescence Search in a third computer agent: quiescencePlayer. This agent acts just like the AlphaBeta agent, except it will deepen its search up to twice for “noisy” nodes. Noisy nodes are identified as those game states where the last placed candy is liable to be stolen on the following turn. QuiescencePlayer has a default depth of 2 and readily beats AlphaBeta players with search depth 3.

**AlmondJoy**

**Minimax-Minimax:**

$ python candyGame.py -b AlmondJoy.txt -p1 minimax -p2 minimax

Minimax A will take (0, 0)

Minimax B will take (0, 1)

Minimax A will take (0, 2)

Minimax B will take (0, 3)

Minimax A will take (0, 4)

Minimax B will take (0, 5)

Minimax A will take (1, 0)

Minimax B will take (1, 1)

Minimax A will take (1, 2)

Minimax B will take (1, 3)

Minimax A will take (2, 5)

Minimax B will take (2, 1)

Minimax A will take (2, 0)

Minimax B will take (2, 2)

Minimax A will take (2, 3)

Minimax B will take (1, 5)

Minimax A will take (3, 0)

Minimax B will take (3, 1)

Minimax A will take (3, 2)

Minimax B will take (3, 3)

Minimax A will take (3, 4)

Minimax B will take (3, 5)

Minimax A will take (5, 2)

Minimax B will take (4, 0)

Minimax A will take (4, 1)

Minimax B will take (4, 2)

Minimax A will take (5, 0)

Minimax B will take (4, 3)

Minimax A will take (5, 3)

Minimax B will take (4, 4)

Minimax A will take (5, 4)

Minimax B will take (4, 5)

Minimax A will take (5, 5)

Minimax B will take (5, 1)

Minimax A will take (1, 4)

Minimax B will take (2, 4)

Game Over!

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)']

['A(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)']

['B(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)', 'A(1)']

['A(1)', 'B(1)', 'A(1)', 'A(1)', 'A(1)', 'A(1)']

Player A : 15 points, 217740 nodes in 88.1 total seconds and 18 moves.

Player B : 21 points, 194736 nodes in 83.3 total seconds and 18 moves.

Player B wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b AlmondJoy.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (0, 0)

AlphaBeta B will take (0, 1)

AlphaBeta A will take (0, 2)

AlphaBeta B will take (0, 3)

AlphaBeta A will take (0, 4)

AlphaBeta B will take (0, 5)

AlphaBeta A will take (1, 1)

AlphaBeta B will take (1, 0)

AlphaBeta A will take (1, 3)

AlphaBeta B will take (1, 2)

AlphaBeta A will take (1, 5)

AlphaBeta B will take (1, 4)

AlphaBeta A will take (2, 0)

AlphaBeta B will take (2, 1)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (2, 3)

AlphaBeta A will take (2, 4)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (3, 1)

AlphaBeta B will take (3, 0)

AlphaBeta A will take (3, 3)

AlphaBeta B will take (3, 2)

AlphaBeta A will take (3, 5)

AlphaBeta B will take (3, 4)

AlphaBeta A will take (4, 0)

AlphaBeta B will take (4, 1)

AlphaBeta A will take (4, 2)

AlphaBeta B will take (4, 3)

AlphaBeta A will take (4, 4)

AlphaBeta B will take (4, 5)

AlphaBeta A will take (5, 1)

AlphaBeta B will take (5, 0)

AlphaBeta A will take (5, 3)

AlphaBeta B will take (5, 2)

AlphaBeta A will take (5, 4)

AlphaBeta B will take (5, 5)

Game Over!

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)']

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)']

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)']

Player A : 17 points, 524874 nodes in 261.6 total seconds and 18 moves.

Player B : 19 points, 428028 nodes in 227.0 total seconds and 18 moves.

Player B wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b AlmondJoy.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (0, 0)

Minimax B will take (0, 1)

AlphaBeta A will take (0, 2)

Minimax B will take (0, 3)

AlphaBeta A will take (0, 4)

Minimax B will take (0, 5)

AlphaBeta A will take (1, 1)

Minimax B will take (1, 0)

AlphaBeta A will take (1, 3)

Minimax B will take (1, 2)

AlphaBeta A will take (1, 5)

Minimax B will take (1, 4)

AlphaBeta A will take (2, 0)

Minimax B will take (2, 1)

AlphaBeta A will take (2, 2)

Minimax B will take (2, 3)

AlphaBeta A will take (2, 4)

Minimax B will take (2, 5)

AlphaBeta A will take (3, 1)

Minimax B will take (3, 0)

AlphaBeta A will take (3, 3)

Minimax B will take (3, 2)

AlphaBeta A will take (3, 5)

Minimax B will take (3, 4)

AlphaBeta A will take (4, 0)

Minimax B will take (4, 1)

AlphaBeta A will take (4, 2)

Minimax B will take (4, 3)

AlphaBeta A will take (4, 4)

Minimax B will take (4, 5)

AlphaBeta A will take (5, 1)

Minimax B will take (5, 0)

AlphaBeta A will take (5, 3)

Minimax B will take (5, 2)

AlphaBeta A will take (5, 4)

Minimax B will take (5, 5)

Game Over!

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)']

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)']

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)']

Player A : 17 points, 524874 nodes in 218.4 total seconds and 18 moves.

Player B : 19 points, 194736 nodes in 88.1 total seconds and 18 moves.

Player B wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b AlmondJoy.txt -p1 minimax -p2 alphabeta

Minimax A will take (0, 0)

AlphaBeta B will take (0, 1)

Minimax A will take (0, 2)

AlphaBeta B will take (0, 3)

Minimax A will take (0, 4)

AlphaBeta B will take (0, 5)

Minimax A will take (1, 0)

AlphaBeta B will take (1, 1)

Minimax A will take (1, 2)

AlphaBeta B will take (1, 3)

Minimax A will take (2, 5)

AlphaBeta B will take (1, 4)

Minimax A will take (1, 5)

AlphaBeta B will take (2, 1)

Minimax A will take (2, 0)

AlphaBeta B will take (2, 2)

Minimax A will take (2, 3)

AlphaBeta B will take (2, 4)

Minimax A will take (3, 1)

AlphaBeta B will take (3, 2)

Minimax A will take (3, 3)

AlphaBeta B will take (3, 4)

Minimax A will take (3, 0)

AlphaBeta B will take (3, 5)

Minimax A will take (4, 0)

AlphaBeta B will take (4, 2)

Minimax A will take (4, 1)

AlphaBeta B will take (4, 3)

Minimax A will take (5, 0)

AlphaBeta B will take (5, 1)

Minimax A will take (4, 4)

AlphaBeta B will take (4, 5)

Minimax A will take (5, 2)

AlphaBeta B will take (5, 3)

Minimax A will take (5, 4)

AlphaBeta B will take (5, 5)

Game Over!

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)', 'A(1)']

['B(1)', 'A(1)', 'B(1)', 'B(1)', 'A(1)', 'A(1)']

['A(1)', 'B(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)']

['A(1)', 'A(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)']

['A(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)']

['A(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)']

Player A : 14 points, 217740 nodes in 89.9 total seconds and 18 moves.

Player B : 22 points, 374922 nodes in 156.3 total seconds and 18 moves.

Player B wins!

**Ayds**

**Minimax-Minimax:**

$ python candyGame.py -b Ayds.txt -p1 minimax -p2 minimax

Minimax A will take (0, 0)

Minimax B will take (0, 4)

Minimax A will take (0, 2)

Minimax B will take (0, 3)

Minimax A will take (0, 1)

Minimax B will take (1, 3)

Minimax A will take (1, 2)

Minimax B will take (1, 4)

Minimax A will take (1, 1)

Minimax B will take (1, 5)

Minimax A will take (2, 2)

Minimax B will take (2, 0)

Minimax A will take (1, 0)

Minimax B will take (2, 3)

Minimax A will take (2, 1)

Minimax B will take (2, 4)

Minimax A will take (3, 3)

Minimax B will take (3, 1)

Minimax A will take (3, 0)

Minimax B will take (3, 4)

Minimax A will take (3, 2)

Minimax B will take (3, 5)

Minimax A will take (5, 3)

Minimax B will take (4, 4)

Minimax A will take (4, 3)

Minimax B will take (4, 5)

Minimax A will take (5, 4)

Minimax B will take (5, 5)

Minimax A will take (4, 0)

Minimax B will take (0, 5)

Minimax A will take (4, 2)

Minimax B will take (2, 5)

Minimax A will take (4, 1)

Minimax B will take (5, 0)

Minimax A will take (5, 1)

Minimax B will take (5, 2)

Game Over!

['A(99)', 'A(1)', 'A(99)', 'B(1)', 'B(99)', 'B(1)']

['A(1)', 'A(99)', 'A(1)', 'B(99)', 'B(1)', 'B(99)']

['A(99)', 'A(1)', 'A(99)', 'B(1)', 'B(99)', 'B(1)']

['A(1)', 'A(99)', 'A(1)', 'A(99)', 'B(1)', 'B(99)']

['A(99)', 'A(1)', 'A(99)', 'A(1)', 'A(99)', 'B(1)']

['A(1)', 'A(99)', 'B(1)', 'A(99)', 'B(1)', 'B(99)']

Player A : 1197 points, 217740 nodes in 102.1 total seconds and 18 moves.

Player B : 603 points, 194736 nodes in 96.4 total seconds and 18 moves.

Player A wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b Ayds.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (0, 0)

AlphaBeta B will take (1, 5)

AlphaBeta A will take (0, 2)

AlphaBeta B will take (0, 4)

AlphaBeta A will take (0, 3)

AlphaBeta B will take (1, 4)

AlphaBeta A will take (1, 3)

AlphaBeta B will take (2, 4)

AlphaBeta A will take (2, 3)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (1, 1)

AlphaBeta B will take (3, 5)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (3, 3)

AlphaBeta A will take (2, 0)

AlphaBeta B will take (4, 4)

AlphaBeta A will take (3, 2)

AlphaBeta B will take (3, 4)

AlphaBeta A will take (3, 1)

AlphaBeta B will take (4, 2)

AlphaBeta A will take (4, 0)

AlphaBeta B will take (4, 1)

AlphaBeta A will take (3, 0)

AlphaBeta B will take (5, 0)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (5, 1)

AlphaBeta A will take (0, 5)

AlphaBeta B will take (5, 3)

AlphaBeta A will take (1, 0)

AlphaBeta B will take (1, 2)

AlphaBeta A will take (2, 1)

AlphaBeta B will take (4, 3)

AlphaBeta A will take (4, 5)

AlphaBeta B will take (5, 2)

AlphaBeta A will take (5, 4)

AlphaBeta B will take (5, 5)

Game Over!

['A(99)', 'A(1)', 'A(99)', 'A(1)', 'B(99)', 'A(1)']

['A(1)', 'A(99)', 'B(1)', 'A(99)', 'A(1)', 'B(99)']

['A(99)', 'A(1)', 'A(99)', 'A(1)', 'B(99)', 'B(1)']

['A(1)', 'A(99)', 'A(1)', 'B(99)', 'B(1)', 'B(99)']

['A(99)', 'B(1)', 'B(99)', 'B(1)', 'B(99)', 'A(1)']

['B(1)', 'B(99)', 'B(1)', 'B(99)', 'A(1)', 'B(99)']

Player A : 803 points, 622536 nodes in 278.9 total seconds and 18 moves.

Player B : 997 points, 687009 nodes in 315.1 total seconds and 18 moves.

Player B wins!

**Alphabeta-Minimax**

$ python candyGame.py -b Ayds.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (0, 0)

Minimax B will take (0, 4)

AlphaBeta A will take (0, 2)

Minimax B will take (0, 3)

AlphaBeta A will take (0, 1)

Minimax B will take (1, 3)

AlphaBeta A will take (1, 2)

Minimax B will take (1, 4)

AlphaBeta A will take (1, 1)

Minimax B will take (1, 5)

AlphaBeta A will take (2, 0)

Minimax B will take (2, 2)

AlphaBeta A will take (2, 1)

Minimax B will take (2, 3)

AlphaBeta A will take (3, 1)

Minimax B will take (3, 2)

AlphaBeta A will take (3, 0)

Minimax B will take (2, 4)

AlphaBeta A will take (4, 2)

Minimax B will take (3, 3)

AlphaBeta A will take (4, 3)

Minimax B will take (3, 4)

AlphaBeta A will take (4, 4)

Minimax B will take (3, 5)

AlphaBeta A will take (4, 5)

Minimax B will take (2, 5)

AlphaBeta A will take (4, 0)

Minimax B will take (0, 5)

AlphaBeta A will take (5, 1)

Minimax B will take (1, 0)

AlphaBeta A will take (5, 3)

Minimax B will take (4, 1)

AlphaBeta A will take (5, 0)

Minimax B will take (5, 2)

AlphaBeta A will take (5, 4)

Minimax B will take (5, 5)

Game Over!

['A(99)', 'A(1)', 'A(99)', 'B(1)', 'B(99)', 'B(1)']

['B(1)', 'A(99)', 'A(1)', 'B(99)', 'B(1)', 'B(99)']

['A(99)', 'A(1)', 'B(99)', 'B(1)', 'B(99)', 'B(1)']

['A(1)', 'A(99)', 'B(1)', 'B(99)', 'A(1)', 'B(99)']

['A(99)', 'B(1)', 'A(99)', 'A(1)', 'A(99)', 'A(1)']

['A(1)', 'A(99)', 'B(1)', 'A(99)', 'A(1)', 'B(99)']

Player A : 999 points, 529986 nodes in 230.7 total seconds and 18 moves.

Player B : 801 points, 194736 nodes in 82.9 total seconds and 18 moves.

Player A wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b Ayds.txt -p1 minimax -p2 alphabeta

Minimax A will take (0, 0)

AlphaBeta B will take (1, 5)

Minimax A will take (0, 2)

AlphaBeta B will take (0, 4)

Minimax A will take (0, 3)

AlphaBeta B will take (1, 4)

Minimax A will take (1, 1)

AlphaBeta B will take (3, 5)

Minimax A will take (1, 3)

AlphaBeta B will take (2, 4)

Minimax A will take (2, 0)

AlphaBeta B will take (4, 2)

Minimax A will take (2, 2)

AlphaBeta B will take (2, 3)

Minimax A will take (1, 2)

AlphaBeta B will take (3, 2)

Minimax A will take (2, 1)

AlphaBeta B will take (3, 1)

Minimax A will take (3, 0)

AlphaBeta B will take (4, 1)

Minimax A will take (4, 0)

AlphaBeta B will take (3, 3)

Minimax A will take (5, 1)

AlphaBeta B will take (5, 2)

Minimax A will take (5, 0)

AlphaBeta B will take (4, 4)

Minimax A will take (0, 1)

AlphaBeta B will take (5, 3)

Minimax A will take (0, 5)

AlphaBeta B will take (1, 0)

Minimax A will take (2, 5)

AlphaBeta B will take (3, 4)

Minimax A will take (4, 3)

AlphaBeta B will take (4, 5)

Minimax A will take (5, 4)

AlphaBeta B will take (5, 5)

Game Over!

['A(99)', 'A(1)', 'A(99)', 'A(1)', 'B(99)', 'A(1)']

['B(1)', 'A(99)', 'A(1)', 'A(99)', 'A(1)', 'B(99)']

['A(99)', 'B(1)', 'A(99)', 'B(1)', 'B(99)', 'A(1)']

['A(1)', 'B(99)', 'B(1)', 'B(99)', 'B(1)', 'B(99)']

['A(99)', 'A(1)', 'B(99)', 'A(1)', 'B(99)', 'B(1)']

['A(1)', 'A(99)', 'B(1)', 'B(99)', 'B(1)', 'B(99)']

Player A : 802 points, 217740 nodes in 106.9 total seconds and 18 moves.

Player B : 998 points, 660043 nodes in 358.7 total seconds and 18 moves.

Player B wins!

**Bit-O-Honey**

**Minimax-Minimax:**

$ python candyGame.py -b Bit-O-Honey.txt -p1 minimax -p2 minimax

Minimax A will take (5, 0)

Minimax B will take (5, 1)

Minimax A will take (3, 1)

Minimax B will take (3, 0)

Minimax A will take (5, 2)

Minimax B will take (5, 3)

Minimax A will take (5, 4)

Minimax B will take (5, 5)

Minimax A will take (4, 3)

Minimax B will take (3, 2)

Minimax A will take (3, 3)

Minimax B will take (4, 4)

Minimax A will take (3, 4)

Minimax B will take (4, 5)

Minimax A will take (3, 5)

Minimax B will take (2, 0)

Minimax A will take (2, 1)

Minimax B will take (2, 2)

Minimax A will take (2, 3)

Minimax B will take (2, 4)

Minimax A will take (2, 5)

Minimax B will take (1, 0)

Minimax A will take (1, 1)

Minimax B will take (1, 2)

Minimax A will take (1, 3)

Minimax B will take (1, 4)

Minimax A will take (1, 5)

Minimax B will take (0, 0)

Minimax A will take (0, 1)

Minimax B will take (0, 2)

Minimax A will take (0, 3)

Minimax B will take (0, 4)

Minimax A will take (0, 5)

Minimax B will take (4, 2)

Minimax A will take (4, 0)

Minimax B will take (4, 1)

Game Over!

['A(1)', 'A(1)', 'A(1)', 'A(1)', 'A(1)', 'A(1)']

['A(2)', 'A(2)', 'A(2)', 'A(2)', 'A(2)', 'A(2)']

['A(4)', 'A(4)', 'A(4)', 'A(4)', 'A(4)', 'A(4)']

['A(8)', 'B(8)', 'A(8)', 'A(8)', 'A(8)', 'A(8)']

['B(16)', 'B(16)', 'B(16)', 'A(16)', 'B(16)', 'A(16)']

['A(32)', 'B(32)', 'A(32)', 'B(32)', 'A(32)', 'B(32)']

Player A : 210 points, 217740 nodes in 90.8 total seconds and 18 moves.

Player B : 168 points, 194736 nodes in 84.5 total seconds and 18 moves.

Player A wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b Bit-O-Honey.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (5, 0)

AlphaBeta B will take (5, 3)

AlphaBeta A will take (5, 2)

AlphaBeta B will take (5, 1)

AlphaBeta A will take (5, 4)

AlphaBeta B will take (5, 5)

AlphaBeta A will take (4, 1)

AlphaBeta B will take (4, 4)

AlphaBeta A will take (4, 0)

AlphaBeta B will take (4, 2)

AlphaBeta A will take (4, 3)

AlphaBeta B will take (4, 5)

AlphaBeta A will take (3, 2)

AlphaBeta B will take (3, 3)

AlphaBeta A will take (2, 0)

AlphaBeta B will take (2, 4)

AlphaBeta A will take (3, 0)

AlphaBeta B will take (3, 4)

AlphaBeta A will take (2, 3)

AlphaBeta B will take (3, 5)

AlphaBeta A will take (3, 1)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (1, 2)

AlphaBeta B will take (1, 3)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (1, 0)

AlphaBeta A will take (1, 1)

AlphaBeta B will take (2, 1)

AlphaBeta A will take (0, 0)

AlphaBeta B will take (1, 4)

AlphaBeta A will take (1, 5)

AlphaBeta B will take (0, 2)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (0, 3)

AlphaBeta A will take (0, 4)

AlphaBeta B will take (0, 5)

Game Over!

['A(1)', 'A(1)', 'B(1)', 'B(1)', 'A(1)', 'B(1)']

['A(2)', 'A(2)', 'A(2)', 'B(2)', 'B(2)', 'A(2)']

['A(4)', 'B(4)', 'A(4)', 'A(4)', 'B(4)', 'B(4)']

['A(8)', 'A(8)', 'A(8)', 'B(8)', 'B(8)', 'B(8)']

['A(16)', 'A(16)', 'B(16)', 'A(16)', 'B(16)', 'B(16)']

['A(32)', 'B(32)', 'A(32)', 'B(32)', 'A(32)', 'B(32)']

Player A : 191 points, 4610363 nodes in 2143.0 total seconds and 18 moves.

Player B : 187 points, 4097692 nodes in 2060.7 total seconds and 18 moves.

Player A wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b Bit-O-Honey.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (5, 0)

Minimax B will take (5, 1)

AlphaBeta A will take (5, 2)

Minimax B will take (5, 3)

AlphaBeta A will take (5, 4)

Minimax B will take (4, 4)

AlphaBeta A will take (4, 1)

Minimax B will take (4, 0)

AlphaBeta A will take (4, 3)

Minimax B will take (4, 2)

AlphaBeta A will take (3, 4)

Minimax B will take (3, 1)

AlphaBeta A will take (3, 5)

Minimax B will take (3, 2)

AlphaBeta A will take (3, 3)

Minimax B will take (3, 0)

AlphaBeta A will take (2, 3)

Minimax B will take (2, 0)

AlphaBeta A will take (2, 1)

Minimax B will take (2, 2)

AlphaBeta A will take (2, 4)

Minimax B will take (2, 5)

AlphaBeta A will take (1, 0)

Minimax B will take (1, 1)

AlphaBeta A will take (1, 2)

Minimax B will take (1, 3)

AlphaBeta A will take (1, 4)

Minimax B will take (1, 5)

AlphaBeta A will take (0, 1)

Minimax B will take (0, 0)

AlphaBeta A will take (0, 2)

Minimax B will take (0, 3)

AlphaBeta A will take (0, 4)

Minimax B will take (0, 5)

AlphaBeta A will take (4, 5)

Minimax B will take (5, 5)

Game Over!

['B(1)', 'A(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)']

['A(2)', 'B(2)', 'A(2)', 'A(2)', 'B(2)', 'B(2)']

['B(4)', 'A(4)', 'B(4)', 'A(4)', 'A(4)', 'B(4)']

['B(8)', 'B(8)', 'A(8)', 'A(8)', 'A(8)', 'A(8)']

['B(16)', 'A(16)', 'B(16)', 'A(16)', 'A(16)', 'A(16)']

['A(32)', 'B(32)', 'A(32)', 'B(32)', 'A(32)', 'B(32)']

Player A : 212 points, 4865132 nodes in 2285.3 total seconds and 18 moves.

Player B : 166 points, 194736 nodes in 91.9 total seconds and 18 moves.

Player A wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b Bit-O-Honey.txt -p1 minimax -p2 alphabeta

Minimax A will take (5, 0)

AlphaBeta B will take (5, 3)

Minimax A will take (5, 2)

AlphaBeta B will take (5, 1)

Minimax A will take (3, 1)

AlphaBeta B will take (4, 1)

Minimax A will take (4, 0)

AlphaBeta B will take (3, 0)

Minimax A will take (3, 3)

AlphaBeta B will take (3, 2)

Minimax A will take (4, 2)

AlphaBeta B will take (4, 3)

Minimax A will take (3, 5)

AlphaBeta B will take (3, 4)

Minimax A will take (2, 0)

AlphaBeta B will take (5, 4)

Minimax A will take (2, 1)

AlphaBeta B will take (4, 4)

Minimax A will take (2, 2)

AlphaBeta B will take (2, 3)

Minimax A will take (2, 5)

AlphaBeta B will take (2, 4)

Minimax A will take (1, 0)

AlphaBeta B will take (1, 1)

Minimax A will take (1, 2)

AlphaBeta B will take (1, 3)

Minimax A will take (1, 4)

AlphaBeta B will take (1, 5)

Minimax A will take (0, 0)

AlphaBeta B will take (0, 1)

Minimax A will take (0, 2)

AlphaBeta B will take (0, 3)

Minimax A will take (0, 4)

AlphaBeta B will take (0, 5)

Minimax A will take (4, 5)

AlphaBeta B will take (5, 5)

Game Over!

['B(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)']

['A(2)', 'B(2)', 'B(2)', 'B(2)', 'B(2)', 'B(2)']

['A(4)', 'A(4)', 'B(4)', 'B(4)', 'B(4)', 'B(4)']

['B(8)', 'A(8)', 'A(8)', 'B(8)', 'B(8)', 'B(8)']

['B(16)', 'A(16)', 'B(16)', 'B(16)', 'B(16)', 'B(16)']

['A(32)', 'B(32)', 'A(32)', 'B(32)', 'B(32)', 'B(32)']

Player A : 106 points, 217740 nodes in 105.9 total seconds and 18 moves.

Player B : 272 points, 3894960 nodes in 2033.7 total seconds and 18 moves.

Player B wins!

**Mounds**

**Minimax-Minimax:**

$ python candyGame.py -b Mounds.txt -p1 minimax -p2 minimax

Minimax A will take (1, 2)

Minimax B will take (1, 3)

Minimax A will take (2, 1)

Minimax B will take (2, 4)

Minimax A will take (3, 1)

Minimax B will take (3, 3)

Minimax A will take (0, 4)

Minimax B will take (3, 2)

Minimax A will take (2, 2)

Minimax B will take (0, 3)

Minimax A will take (1, 5)

Minimax B will take (0, 2)

Minimax A will take (1, 1)

Minimax B will take (0, 1)

Minimax A will take (3, 4)

Minimax B will take (4, 2)

Minimax A will take (4, 3)

Minimax B will take (4, 1)

Minimax A will take (4, 4)

Minimax B will take (0, 0)

Minimax A will take (0, 5)

Minimax B will take (2, 5)

Minimax A will take (3, 5)

Minimax B will take (1, 0)

Minimax A will take (2, 0)

Minimax B will take (3, 0)

Minimax A will take (4, 0)

Minimax B will take (4, 5)

Minimax A will take (5, 0)

Minimax B will take (5, 1)

Minimax A will take (5, 2)

Minimax B will take (5, 3)

Minimax A will take (5, 4)

Minimax B will take (5, 5)

Minimax A will take (1, 4)

Minimax B will take (2, 3)

Game Over!

['B(1)', 'B(1)', 'B(1)', 'B(1)', 'A(1)', 'A(1)']

['A(1)', 'B(3)', 'A(4)', 'B(4)', 'A(3)', 'A(1)']

['B(1)', 'A(4)', 'B(2)', 'B(2)', 'B(4)', 'A(1)']

['B(1)', 'B(4)', 'A(2)', 'B(2)', 'A(4)', 'A(1)']

['A(1)', 'B(3)', 'B(4)', 'A(4)', 'A(3)', 'B(1)']

['B(1)', 'B(1)', 'A(1)', 'A(1)', 'B(1)', 'B(1)']

Player A : 33 points, 217740 nodes in 133.2 total seconds and 18 moves.

Player B : 39 points, 194736 nodes in 126.7 total seconds and 18 moves.

Player B wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b Mounds.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (1, 2)

AlphaBeta B will take (1, 3)

AlphaBeta A will take (2, 1)

AlphaBeta B will take (2, 4)

AlphaBeta A will take (3, 1)

AlphaBeta B will take (4, 3)

AlphaBeta A will take (4, 2)

AlphaBeta B will take (3, 4)

AlphaBeta A will take (1, 1)

AlphaBeta B will take (1, 4)

AlphaBeta A will take (4, 1)

AlphaBeta B will take (4, 4)

AlphaBeta A will take (0, 0)

AlphaBeta B will take (0, 2)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (0, 3)

AlphaBeta A will take (0, 4)

AlphaBeta B will take (0, 5)

AlphaBeta A will take (1, 0)

AlphaBeta B will take (1, 5)

AlphaBeta A will take (2, 0)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (3, 0)

AlphaBeta B will take (3, 5)

AlphaBeta A will take (4, 0)

AlphaBeta B will take (4, 5)

AlphaBeta A will take (5, 0)

AlphaBeta B will take (5, 1)

AlphaBeta A will take (5, 2)

AlphaBeta B will take (5, 3)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (2, 3)

AlphaBeta A will take (3, 2)

AlphaBeta B will take (3, 3)

AlphaBeta A will take (5, 4)

AlphaBeta B will take (5, 5)

Game Over!

['A(1)', 'A(1)', 'B(1)', 'B(1)', 'A(1)', 'B(1)']

['A(1)', 'A(3)', 'A(4)', 'B(4)', 'B(3)', 'B(1)']

['A(1)', 'A(4)', 'A(2)', 'B(2)', 'B(4)', 'B(1)']

['A(1)', 'A(4)', 'B(2)', 'B(2)', 'B(4)', 'B(1)']

['A(1)', 'A(3)', 'A(4)', 'B(4)', 'B(3)', 'B(1)']

['A(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)']

Player A : 33 points, 1187759 nodes in 537.8 total seconds and 18 moves.

Player B : 39 points, 1093397 nodes in 519.0 total seconds and 18 moves.

Player B wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b Mounds.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (1, 2)

Minimax B will take (1, 3)

AlphaBeta A will take (2, 1)

Minimax B will take (2, 4)

AlphaBeta A will take (3, 1)

Minimax B will take (3, 3)

AlphaBeta A will take (3, 2)

Minimax B will take (3, 4)

AlphaBeta A will take (4, 1)

Minimax B will take (4, 2)

AlphaBeta A will take (1, 1)

Minimax B will take (1, 4)

AlphaBeta A will take (0, 0)

Minimax B will take (2, 2)

AlphaBeta A will take (2, 3)

Minimax B will take (0, 4)

AlphaBeta A will take (0, 1)

Minimax B will take (0, 2)

AlphaBeta A will take (0, 3)

Minimax B will take (0, 5)

AlphaBeta A will take (1, 0)

Minimax B will take (1, 5)

AlphaBeta A will take (2, 0)

Minimax B will take (2, 5)

AlphaBeta A will take (3, 0)

Minimax B will take (3, 5)

AlphaBeta A will take (4, 5)

Minimax B will take (4, 4)

AlphaBeta A will take (4, 3)

Minimax B will take (4, 0)

AlphaBeta A will take (5, 0)

Minimax B will take (5, 1)

AlphaBeta A will take (5, 2)

Minimax B will take (5, 3)

AlphaBeta A will take (5, 4)

Minimax B will take (5, 5)

Game Over!

['A(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)']

['A(1)', 'A(3)', 'A(4)', 'B(4)', 'B(3)', 'B(1)']

['A(1)', 'A(4)', 'B(2)', 'A(2)', 'B(4)', 'B(1)']

['A(1)', 'A(4)', 'A(2)', 'B(2)', 'B(4)', 'B(1)']

['B(1)', 'A(3)', 'B(4)', 'A(4)', 'B(3)', 'B(1)']

['A(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)']

Player A : 34 points, 1210471 nodes in 605.7 total seconds and 18 moves.

Player B : 38 points, 194736 nodes in 102.1 total seconds and 18 moves.

Player B wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b Mounds.txt -p1 minimax -p2 alphabeta

Minimax A will take (1, 2)

AlphaBeta B will take (1, 3)

Minimax A will take (2, 1)

AlphaBeta B will take (2, 4)

Minimax A will take (3, 1)

AlphaBeta B will take (4, 3)

Minimax A will take (4, 1)

AlphaBeta B will take (4, 2)

Minimax A will take (3, 2)

AlphaBeta B will take (3, 3)

Minimax A will take (2, 2)

AlphaBeta B will take (2, 3)

Minimax A will take (1, 1)

AlphaBeta B will take (3, 4)

Minimax A will take (0, 0)

AlphaBeta B will take (0, 4)

Minimax A will take (0, 1)

AlphaBeta B will take (0, 2)

Minimax A will take (0, 3)

AlphaBeta B will take (0, 5)

Minimax A will take (1, 0)

AlphaBeta B will take (4, 4)

Minimax A will take (2, 5)

AlphaBeta B will take (3, 5)

Minimax A will take (3, 0)

AlphaBeta B will take (4, 0)

Minimax A will take (2, 0)

AlphaBeta B will take (1, 4)

Minimax A will take (1, 5)

AlphaBeta B will take (4, 5)

Minimax A will take (5, 0)

AlphaBeta B will take (5, 1)

Minimax A will take (5, 2)

AlphaBeta B will take (5, 3)

Minimax A will take (5, 4)

AlphaBeta B will take (5, 5)

Game Over!

['A(1)', 'A(1)', 'B(1)', 'A(1)', 'B(1)', 'B(1)']

['A(1)', 'A(3)', 'A(4)', 'B(4)', 'B(3)', 'A(1)']

['A(1)', 'A(4)', 'B(2)', 'B(2)', 'B(4)', 'B(1)']

['A(1)', 'A(4)', 'A(2)', 'B(2)', 'B(4)', 'B(1)']

['B(1)', 'B(3)', 'A(4)', 'B(4)', 'B(3)', 'B(1)']

['B(1)', 'A(1)', 'B(1)', 'B(1)', 'B(1)', 'B(1)']

Player A : 29 points, 217740 nodes in 109.0 total seconds and 18 moves.

Player B : 43 points, 1188439 nodes in 672.9 total seconds and 18 moves.

Player B wins!

**ReesesPieces**

**Minimax-Minimax:**

$ python candyGame.py -b ReesesPieces.txt -p1 minimax -p2 minimax

Minimax A will take (4, 0)

Minimax B will take (4, 3)

Minimax A will take (5, 0)

Minimax B will take (5, 2)

Minimax A will take (5, 4)

Minimax B will take (4, 4)

Minimax A will take (3, 2)

Minimax B will take (4, 2)

Minimax A will take (5, 1)

Minimax B will take (4, 1)

Minimax A will take (5, 3)

Minimax B will take (3, 3)

Minimax A will take (0, 1)

Minimax B will take (2, 1)

Minimax A will take (2, 3)

Minimax B will take (1, 1)

Minimax A will take (3, 1)

Minimax B will take (3, 0)

Minimax A will take (2, 2)

Minimax B will take (2, 0)

Minimax A will take (1, 2)

Minimax B will take (1, 0)

Minimax A will take (0, 0)

Minimax B will take (0, 2)

Minimax A will take (1, 3)

Minimax B will take (0, 3)

Minimax A will take (2, 5)

Minimax B will take (5, 5)

Minimax A will take (0, 4)

Minimax B will take (0, 5)

Minimax A will take (1, 4)

Minimax B will take (3, 4)

Minimax A will take (3, 5)

Minimax B will take (1, 5)

Minimax A will take (2, 4)

Minimax B will take (4, 5)

Game Over!

['A(66)', 'B(76)', 'B(28)', 'B(66)', 'A(11)', 'B(9)']

['B(31)', 'B(39)', 'A(50)', 'A(8)', 'A(33)', 'B(14)']

['B(80)', 'B(76)', 'A(39)', 'A(59)', 'A(2)', 'A(48)']

['B(50)', 'B(73)', 'A(43)', 'B(3)', 'A(13)', 'B(3)']

['B(99)', 'B(45)', 'B(72)', 'B(87)', 'B(49)', 'B(4)']

['A(80)', 'B(63)', 'A(92)', 'A(28)', 'A(61)', 'B(53)']

Player A : 633 points, 217740 nodes in 126.9 total seconds and 18 moves.

Player B : 1020 points, 194736 nodes in 125.1 total seconds and 18 moves.

Player B wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b ReesesPieces.txt -p1 al-p2 alphabeta

AlphaBeta A will take (2, 0)

AlphaBeta B will take (5, 2)

AlphaBeta A will take (4, 0)

AlphaBeta B will take (4, 3)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (5, 0)

AlphaBeta A will take (2, 1)

AlphaBeta B will take (4, 1)

AlphaBeta A will take (3, 1)

AlphaBeta B will take (5, 1)

AlphaBeta A will take (4, 2)

AlphaBeta B will take (5, 4)

AlphaBeta A will take (0, 0)

AlphaBeta B will take (2, 3)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (3, 3)

AlphaBeta A will take (3, 2)

AlphaBeta B will take (5, 5)

AlphaBeta A will take (1, 2)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (1, 3)

AlphaBeta B will take (2, 4)

AlphaBeta A will take (0, 3)

AlphaBeta B will take (3, 0)

AlphaBeta A will take (1, 4)

AlphaBeta B will take (1, 5)

AlphaBeta A will take (0, 4)

AlphaBeta B will take (0, 5)

AlphaBeta A will take (4, 4)

AlphaBeta B will take (4, 5)

AlphaBeta A will take (3, 4)

AlphaBeta B will take (3, 5)

AlphaBeta A will take (0, 2)

AlphaBeta B will take (1, 0)

AlphaBeta A will take (1, 1)

AlphaBeta B will take (5, 3)

Game Over!

['A(66)', 'A(76)', 'A(28)', 'A(66)', 'B(11)', 'B(9)']

['A(31)', 'A(39)', 'A(50)', 'A(8)', 'A(33)', 'B(14)']

['A(80)', 'A(76)', 'A(39)', 'B(59)', 'A(2)', 'B(48)']

['B(50)', 'A(73)', 'A(43)', 'A(3)', 'B(13)', 'B(3)']

['A(99)', 'B(45)', 'A(72)', 'B(87)', 'A(49)', 'B(4)']

['B(80)', 'B(63)', 'B(92)', 'B(28)', 'B(61)', 'B(53)']

Player A : 933 points, 1063581 nodes in 541.5 total seconds and 18 moves.

Player B : 720 points, 1363347 nodes in 740.3 total seconds and 18 moves.

Player A wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b ReesesPieces.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (2, 0)

Minimax B will take (4, 3)

AlphaBeta A will take (4, 0)

Minimax B will take (5, 0)

AlphaBeta A will take (0, 1)

Minimax B will take (5, 2)

AlphaBeta A will take (2, 1)

Minimax B will take (5, 4)

AlphaBeta A will take (4, 2)

Minimax B will take (5, 5)

AlphaBeta A will take (0, 0)

Minimax B will take (4, 4)

AlphaBeta A will take (3, 1)

Minimax B will take (3, 0)

AlphaBeta A will take (0, 3)

Minimax B will take (2, 5)

AlphaBeta A will take (1, 2)

Minimax B will take (4, 1)

AlphaBeta A will take (5, 1)

Minimax B will take (0, 2)

AlphaBeta A will take (5, 3)

Minimax B will take (0, 5)

AlphaBeta A will take (0, 4)

Minimax B will take (1, 5)

AlphaBeta A will take (1, 4)

Minimax B will take (2, 4)

AlphaBeta A will take (1, 3)

Minimax B will take (3, 2)

AlphaBeta A will take (2, 2)

Minimax B will take (1, 0)

AlphaBeta A will take (1, 1)

Minimax B will take (3, 3)

AlphaBeta A will take (2, 3)

Minimax B will take (3, 4)

AlphaBeta A will take (3, 5)

Minimax B will take (4, 5)

Game Over!

['A(66)', 'A(76)', 'B(28)', 'A(66)', 'A(11)', 'B(9)']

['A(31)', 'A(39)', 'A(50)', 'A(8)', 'A(33)', 'A(14)']

['A(80)', 'A(76)', 'A(39)', 'A(59)', 'B(2)', 'B(48)']

['B(50)', 'A(73)', 'B(43)', 'B(3)', 'B(13)', 'B(3)']

['A(99)', 'B(45)', 'A(72)', 'B(87)', 'B(49)', 'B(4)']

['B(80)', 'A(63)', 'B(92)', 'A(28)', 'B(61)', 'B(53)']

Player A : 983 points, 1083505 nodes in 549.8 total seconds and 18 moves.

Player B : 670 points, 194736 nodes in 98.2 total seconds and 18 moves.

Player A wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b ReesesPieces.txt -p1 minimax -p2 alphabeta

Minimax A will take (4, 0)

AlphaBeta B will take (5, 2)

Minimax A will take (3, 2)

AlphaBeta B will take (4, 2)

Minimax A will take (4, 1)

AlphaBeta B will take (3, 1)

Minimax A will take (3, 0)

AlphaBeta B will take (4, 3)

Minimax A will take (2, 1)

AlphaBeta B will take (2, 2)

Minimax A will take (2, 0)

AlphaBeta B will take (5, 0)

Minimax A will take (0, 1)

AlphaBeta B will take (5, 1)

Minimax A will take (0, 0)

AlphaBeta B will take (5, 4)

Minimax A will take (0, 3)

AlphaBeta B will take (5, 5)

Minimax A will take (2, 4)

AlphaBeta B will take (2, 3)

Minimax A will take (1, 3)

AlphaBeta B will take (3, 3)

Minimax A will take (1, 2)

AlphaBeta B will take (2, 5)

Minimax A will take (0, 5)

AlphaBeta B will take (1, 5)

Minimax A will take (1, 1)

AlphaBeta B will take (0, 2)

Minimax A will take (4, 5)

AlphaBeta B will take (4, 4)

Minimax A will take (0, 4)

AlphaBeta B will take (1, 4)

Minimax A will take (1, 0)

AlphaBeta B will take (5, 3)

Minimax A will take (3, 4)

AlphaBeta B will take (3, 5)

Game Over!

['A(66)', 'A(76)', 'B(28)', 'A(66)', 'B(11)', 'A(9)']

['A(31)', 'A(39)', 'A(50)', 'B(8)', 'B(33)', 'B(14)']

['A(80)', 'A(76)', 'A(39)', 'B(59)', 'B(2)', 'B(48)']

['A(50)', 'A(73)', 'B(43)', 'B(3)', 'B(13)', 'B(3)']

['A(99)', 'B(45)', 'A(72)', 'B(87)', 'B(49)', 'B(4)']

['B(80)', 'B(63)', 'B(92)', 'B(28)', 'B(61)', 'B(53)']

Player A : 826 points, 217740 nodes in 102.3 total seconds and 18 moves.

Player B : 827 points, 1544871 nodes in 770.9 total seconds and 18 moves.

Player B wins!

**Oases**

**Minimax-Minimax:**

$ python candyGame.py -b oases.txt -p1 minimax -p2 minimax

Minimax A will take (1, 4)

Minimax B will take (0, 3)

Minimax A will take (0, 4)

Minimax B will take (0, 5)

Minimax A will take (4, 1)

Minimax B will take (1, 2)

Minimax A will take (0, 2)

Minimax B will take (2, 1)

Minimax A will take (2, 2)

Minimax B will take (2, 3)

Minimax A will take (1, 3)

Minimax B will take (3, 0)

Minimax A will take (2, 0)

Minimax B will take (3, 2)

Minimax A will take (3, 3)

Minimax B will take (3, 4)

Minimax A will take (0, 0)

Minimax B will take (5, 5)

Minimax A will take (3, 5)

Minimax B will take (4, 5)

Minimax A will take (4, 4)

Minimax B will take (5, 4)

Minimax A will take (0, 1)

Minimax B will take (4, 3)

Minimax A will take (1, 0)

Minimax B will take (1, 1)

Minimax A will take (1, 5)

Minimax B will take (2, 4)

Minimax A will take (2, 5)

Minimax B will take (5, 0)

Minimax A will take (3, 1)

Minimax B will take (4, 0)

Minimax A will take (4, 2)

Minimax B will take (5, 1)

Minimax A will take (5, 2)

Minimax B will take (5, 3)

Game Over!

['A(4)', 'A(0)', 'A(0)', 'A(0)', 'A(0)', 'A(0)']

['A(0)', 'B(0)', 'A(0)', 'A(0)', 'B(10)', 'A(0)']

['A(0)', 'A(0)', 'A(6)', 'B(0)', 'A(0)', 'A(0)']

['B(0)', 'A(0)', 'A(0)', 'B(6)', 'B(0)', 'A(0)']

['B(0)', 'B(10)', 'A(0)', 'B(0)', 'B(0)', 'B(0)']

['B(0)', 'A(0)', 'B(0)', 'B(0)', 'B(0)', 'B(4)']

Player A : 10 points, 217740 nodes in 115.2 total seconds and 18 moves.

Player B : 30 points, 194736 nodes in 107.9 total seconds and 18 moves.

Player B wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b oases.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (1, 4)

AlphaBeta B will take (0, 3)

AlphaBeta A will take (0, 4)

AlphaBeta B will take (1, 2)

AlphaBeta A will take (0, 2)

AlphaBeta B will take (4, 1)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (2, 3)

AlphaBeta A will take (1, 3)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (1, 5)

AlphaBeta B will take (5, 5)

AlphaBeta A will take (3, 3)

AlphaBeta B will take (0, 0)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (0, 5)

AlphaBeta A will take (1, 0)

AlphaBeta B will take (1, 1)

AlphaBeta A will take (2, 1)

AlphaBeta B will take (3, 1)

AlphaBeta A will take (3, 2)

AlphaBeta B will take (4, 2)

AlphaBeta A will take (3, 0)

AlphaBeta B will take (2, 0)

AlphaBeta A will take (4, 3)

AlphaBeta B will take (2, 4)

AlphaBeta A will take (3, 4)

AlphaBeta B will take (3, 5)

AlphaBeta A will take (4, 0)

AlphaBeta B will take (4, 4)

AlphaBeta A will take (4, 5)

AlphaBeta B will take (5, 0)

AlphaBeta A will take (5, 1)

AlphaBeta B will take (5, 2)

AlphaBeta A will take (5, 3)

AlphaBeta B will take (5, 4)

Game Over!

['A(4)', 'A(0)', 'A(0)', 'A(0)', 'A(0)', 'B(0)']

['B(0)', 'A(0)', 'A(0)', 'A(0)', 'A(10)', 'A(0)']

['B(0)', 'B(0)', 'A(6)', 'A(0)', 'A(0)', 'A(0)']

['B(0)', 'A(0)', 'B(0)', 'A(6)', 'A(0)', 'B(0)']

['A(0)', 'B(10)', 'A(0)', 'A(0)', 'B(0)', 'A(0)']

['B(0)', 'A(0)', 'A(0)', 'B(0)', 'B(0)', 'B(4)']

Player A : 26 points, 577443 nodes in 295.8 total seconds and 18 moves.

Player B : 14 points, 702201 nodes in 374.3 total seconds and 18 moves.

Player A wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b oases.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (1, 4)

Minimax B will take (0, 3)

AlphaBeta A will take (0, 4)

Minimax B will take (0, 5)

AlphaBeta A will take (4, 1)

Minimax B will take (1, 2)

AlphaBeta A will take (0, 2)

Minimax B will take (2, 1)

AlphaBeta A will take (2, 2)

Minimax B will take (2, 3)

AlphaBeta A will take (1, 3)

Minimax B will take (3, 0)

AlphaBeta A will take (2, 0)

Minimax B will take (3, 2)

AlphaBeta A will take (3, 3)

Minimax B will take (3, 4)

AlphaBeta A will take (5, 5)

Minimax B will take (0, 0)

AlphaBeta A will take (0, 1)

Minimax B will take (1, 0)

AlphaBeta A will take (1, 1)

Minimax B will take (2, 4)

AlphaBeta A will take (1, 5)

Minimax B will take (2, 5)

AlphaBeta A will take (3, 1)

Minimax B will take (3, 5)

AlphaBeta A will take (4, 5)

Minimax B will take (4, 0)

AlphaBeta A will take (4, 2)

Minimax B will take (4, 3)

AlphaBeta A will take (4, 4)

Minimax B will take (5, 0)

AlphaBeta A will take (5, 1)

Minimax B will take (5, 2)

AlphaBeta A will take (5, 3)

Minimax B will take (5, 4)

Game Over!

['A(4)', 'A(0)', 'A(0)', 'A(0)', 'A(0)', 'B(0)']

['A(0)', 'A(0)', 'A(0)', 'A(0)', 'B(10)', 'B(0)']

['A(0)', 'A(0)', 'A(6)', 'B(0)', 'B(0)', 'B(0)']

['A(0)', 'A(0)', 'A(0)', 'A(6)', 'A(0)', 'A(0)']

['B(0)', 'A(10)', 'A(0)', 'A(0)', 'A(0)', 'A(0)']

['A(0)', 'A(0)', 'A(0)', 'A(0)', 'B(0)', 'A(4)']

Player A : 30 points, 544068 nodes in 241.3 total seconds and 18 moves.

Player B : 10 points, 194736 nodes in 100.3 total seconds and 18 moves.

Player A wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b oases.txt -p1 minimax -p2 alphabeta

Minimax A will take (1, 4)

AlphaBeta B will take (0, 3)

Minimax A will take (0, 4)

AlphaBeta B will take (1, 2)

Minimax A will take (0, 2)

AlphaBeta B will take (4, 1)

Minimax A will take (2, 1)

AlphaBeta B will take (3, 1)

Minimax A will take (3, 0)

AlphaBeta B will take (1, 1)

Minimax A will take (1, 3)

AlphaBeta B will take (0, 5)

Minimax A will take (1, 5)

AlphaBeta B will take (2, 5)

Minimax A will take (2, 2)

AlphaBeta B will take (2, 4)

Minimax A will take (4, 0)

AlphaBeta B will take (2, 3)

Minimax A will take (4, 2)

AlphaBeta B will take (3, 2)

Minimax A will take (3, 5)

AlphaBeta B will take (3, 3)

Minimax A will take (5, 5)

AlphaBeta B will take (3, 4)

Minimax A will take (4, 5)

AlphaBeta B will take (0, 0)

Minimax A will take (0, 1)

AlphaBeta B will take (1, 0)

Minimax A will take (2, 0)

AlphaBeta B will take (4, 3)

Minimax A will take (4, 4)

AlphaBeta B will take (5, 0)

Minimax A will take (5, 1)

AlphaBeta B will take (5, 2)

Minimax A will take (5, 3)

AlphaBeta B will take (5, 4)

Game Over!

['A(4)', 'A(0)', 'A(0)', 'A(0)', 'A(0)', 'A(0)']

['A(0)', 'A(0)', 'A(0)', 'B(0)', 'B(10)', 'A(0)']

['A(0)', 'A(0)', 'B(6)', 'B(0)', 'B(0)', 'B(0)']

['A(0)', 'B(0)', 'B(0)', 'B(6)', 'A(0)', 'A(0)']

['A(0)', 'A(10)', 'B(0)', 'A(0)', 'A(0)', 'A(0)']

['A(0)', 'B(0)', 'A(0)', 'A(0)', 'B(0)', 'A(4)']

Player A : 18 points, 217740 nodes in 105.1 total seconds and 18 moves.

Player B : 22 points, 753673 nodes in 381.6 total seconds and 18 moves.

Player B wins!

**Long**

**Minimax-Minimax:**

$ python candyGame.py -b long.txt -p1 minimax -p2 minimax

Minimax A will take (0, 2)

Minimax B will take (0, 5)

Minimax A will take (0, 10)

Minimax B will take (0, 7)

Minimax A will take (1, 8)

Minimax B will take (2, 5)

Minimax A will take (0, 8)

Minimax B will take (0, 6)

Minimax A will take (1, 7)

Minimax B will take (1, 2)

Minimax A will take (2, 7)

Minimax B will take (2, 9)

Minimax A will take (1, 4)

Minimax B will take (0, 4)

Minimax A will take (2, 6)

Minimax B will take (2, 4)

Minimax A will take (1, 6)

Minimax B will take (1, 5)

Minimax A will take (0, 3)

Minimax B will take (1, 9)

Minimax A will take (1, 3)

Minimax B will take (2, 8)

Minimax A will take (0, 9)

Minimax B will take (1, 1)

Minimax A will take (0, 1)

Minimax B will take (2, 1)

Minimax A will take (2, 10)

Minimax B will take (1, 11)

Minimax A will take (1, 10)

Minimax B will take (0, 0)

Minimax A will take (1, 0)

Minimax B will take (2, 0)

Minimax A will take (0, 11)

Minimax B will take (0, 12)

Minimax A will take (1, 12)

Minimax B will take (2, 11)

Minimax A will take (2, 12)

Minimax B will take (2, 2)

Minimax A will take (2, 3)

Game Over!

['A(1)', 'A(1)', 'A(4)', 'A(1)', 'A(2)', 'B(4)', 'A(1)', 'A(4)', 'A(2)', 'A(2)', 'A(4)', 'A(1)', 'A(1)']

['B(1)', 'A(0)', 'B(3)', 'A(0)', 'A(1)', 'B(0)', 'B(3)', 'A(0)', 'B(1)', 'A(0)', 'A(3)', 'A(0)', 'A(1)']

['B(1)', 'B(1)', 'A(4)', 'A(1)', 'A(2)', 'B(4)', 'A(1)', 'B(4)', 'B(2)', 'B(2)', 'A(4)', 'A(1)', 'A(1)']

Player A : 43 points, 299080 nodes in 126.2 total seconds and 20 moves.

Player B : 26 points, 269800 nodes in 119.5 total seconds and 19 moves.

Player A wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b long.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (0, 2)

AlphaBeta B will take (0, 5)

AlphaBeta A will take (0, 10)

AlphaBeta B will take (0, 7)

AlphaBeta A will take (2, 2)

AlphaBeta B will take (2, 7)

AlphaBeta A will take (1, 2)

AlphaBeta B will take (1, 6)

AlphaBeta A will take (2, 10)

AlphaBeta B will take (2, 5)

AlphaBeta A will take (1, 10)

AlphaBeta B will take (0, 9)

AlphaBeta A will take (1, 9)

AlphaBeta B will take (0, 8)

AlphaBeta A will take (1, 8)

AlphaBeta B will take (1, 7)

AlphaBeta A will take (0, 4)

AlphaBeta B will take (2, 9)

AlphaBeta A will take (2, 4)

AlphaBeta B will take (2, 8)

AlphaBeta A will take (1, 4)

AlphaBeta B will take (1, 5)

AlphaBeta A will take (1, 3)

AlphaBeta B will take (0, 3)

AlphaBeta A will take (0, 0)

AlphaBeta B will take (0, 6)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (0, 11)

AlphaBeta A will take (1, 11)

AlphaBeta B will take (2, 3)

AlphaBeta A will take (0, 12)

AlphaBeta B will take (1, 12)

AlphaBeta A will take (1, 0)

AlphaBeta B will take (2, 6)

AlphaBeta A will take (2, 0)

AlphaBeta B will take (1, 1)

AlphaBeta A will take (2, 1)

AlphaBeta B will take (2, 11)

AlphaBeta A will take (2, 12)

Game Over!

['A(1)', 'A(1)', 'A(4)', 'B(1)', 'A(2)', 'B(4)', 'B(1)', 'B(4)', 'A(2)', 'B(2)', 'A(4)', 'A(1)', 'A(1)']

['A(1)', 'A(0)', 'A(3)', 'A(0)', 'A(1)', 'B(0)', 'B(3)', 'B(0)', 'B(1)', 'A(0)', 'A(3)', 'A(0)', 'B(1)']

['A(1)', 'A(1)', 'A(4)', 'B(1)', 'A(2)', 'B(4)', 'B(1)', 'B(4)', 'B(2)', 'B(2)', 'A(4)', 'B(1)', 'A(1)']

Player A : 37 points, 1502260 nodes in 816.5 total seconds and 20 moves.

Player B : 32 points, 1419671 nodes in 792.2 total seconds and 19 moves.

Player A wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b long.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (0, 2)

Minimax B will take (0, 5)

AlphaBeta A will take (0, 10)

Minimax B will take (0, 7)

AlphaBeta A will take (2, 2)

Minimax B will take (2, 5)

AlphaBeta A will take (1, 2)

Minimax B will take (0, 1)

AlphaBeta A will take (1, 1)

Minimax B will take (1, 6)

AlphaBeta A will take (0, 4)

Minimax B will take (0, 3)

AlphaBeta A will take (1, 3)

Minimax B will take (1, 10)

AlphaBeta A will take (2, 10)

Minimax B will take (0, 9)

AlphaBeta A will take (2, 3)

Minimax B will take (2, 4)

AlphaBeta A will take (1, 4)

Minimax B will take (1, 5)

AlphaBeta A will take (0, 8)

Minimax B will take (2, 7)

AlphaBeta A will take (2, 8)

Minimax B will take (0, 6)

AlphaBeta A will take (0, 0)

Minimax B will take (0, 11)

AlphaBeta A will take (0, 12)

Minimax B will take (1, 0)

AlphaBeta A will take (2, 0)

Minimax B will take (1, 8)

AlphaBeta A will take (2, 1)

Minimax B will take (1, 12)

AlphaBeta A will take (2, 6)

Minimax B will take (2, 9)

AlphaBeta A will take (1, 11)

Minimax B will take (1, 7)

AlphaBeta A will take (1, 9)

Minimax B will take (2, 11)

AlphaBeta A will take (2, 12)

Game Over!

['A(1)', 'A(1)', 'A(4)', 'A(1)', 'A(2)', 'B(4)', 'B(1)', 'B(4)', 'A(2)', 'B(2)', 'A(4)', 'B(1)', 'A(1)']

['B(1)', 'A(0)', 'A(3)', 'A(0)', 'B(1)', 'B(0)', 'B(3)', 'B(0)', 'B(1)', 'A(0)', 'B(3)', 'A(0)', 'B(1)']

['A(1)', 'A(1)', 'A(4)', 'B(1)', 'A(2)', 'B(4)', 'A(1)', 'B(4)', 'A(2)', 'B(2)', 'A(4)', 'B(1)', 'A(1)']

Player A : 35 points, 1374885 nodes in 751.5 total seconds and 20 moves.

Player B : 34 points, 269800 nodes in 142.0 total seconds and 19 moves.

Player A wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b long.txt -p1 minimax -p2 alphabeta

Minimax A will take (0, 2)

AlphaBeta B will take (0, 5)

Minimax A will take (0, 10)

AlphaBeta B will take (0, 7)

Minimax A will take (1, 8)

AlphaBeta B will take (0, 8)

Minimax A will take (0, 9)

AlphaBeta B will take (1, 9)

Minimax A will take (1, 10)

AlphaBeta B will take (2, 5)

Minimax A will take (2, 2)

AlphaBeta B will take (1, 6)

Minimax A will take (1, 2)

AlphaBeta B will take (2, 7)

Minimax A will take (0, 4)

AlphaBeta B will take (1, 4)

Minimax A will take (2, 9)

AlphaBeta B will take (2, 8)

Minimax A will take (1, 3)

AlphaBeta B will take (2, 4)

Minimax A will take (2, 10)

AlphaBeta B will take (2, 3)

Minimax A will take (0, 3)

AlphaBeta B will take (0, 6)

Minimax A will take (0, 0)

AlphaBeta B will take (0, 1)

Minimax A will take (1, 1)

AlphaBeta B will take (2, 1)

Minimax A will take (0, 11)

AlphaBeta B will take (1, 0)

Minimax A will take (0, 12)

AlphaBeta B will take (1, 5)

Minimax A will take (1, 12)

AlphaBeta B will take (1, 7)

Minimax A will take (2, 0)

AlphaBeta B will take (1, 11)

Minimax A will take (2, 11)

AlphaBeta B will take (2, 6)

Minimax A will take (2, 12)

Game Over!

['B(1)', 'A(1)', 'A(4)', 'A(1)', 'A(2)', 'B(4)', 'B(1)', 'B(4)', 'A(2)', 'B(2)', 'A(4)', 'A(1)', 'A(1)']

['B(1)', 'B(0)', 'A(3)', 'A(0)', 'B(1)', 'B(0)', 'B(3)', 'B(0)', 'B(1)', 'A(0)', 'A(3)', 'A(0)', 'A(1)']

['A(1)', 'B(1)', 'B(4)', 'B(1)', 'B(2)', 'B(4)', 'B(1)', 'B(4)', 'B(2)', 'A(2)', 'A(4)', 'A(1)', 'A(1)']

Player A : 32 points, 299080 nodes in 159.2 total seconds and 20 moves.

Player B : 37 points, 1458470 nodes in 915.6 total seconds and 19 moves.

Player B wins!

**Small**

**Minimax-Minimax:**

$ python candyGame.py -b small.txt -p1 minimax -p2 minimax

Minimax A will take (1, 1)

Minimax B will take (1, 0)

Minimax A will take (1, 2)

Minimax B will take (0, 0)

Minimax A will take (0, 1)

Minimax B will take (0, 2)

Minimax A will take (2, 0)

Minimax B will take (2, 1)

Minimax A will take (2, 2)

Game Over!

['A(1)', 'A(0)', 'B(1)']

['B(2)', 'A(4)', 'A(2)']

['A(1)', 'A(0)', 'A(1)']

Player A : 9 points, 945 nodes in 0.2 total seconds and 5 moves.

Player B : 3 points, 600 nodes in 0.1 total seconds and 4 moves.

Player A wins!

**Alphabeta-Alphabeta:**

$ python candyGame.py -b small.txt -p1 alphabeta -p2 alphabeta

AlphaBeta A will take (1, 1)

AlphaBeta B will take (1, 0)

AlphaBeta A will take (1, 2)

AlphaBeta B will take (0, 0)

AlphaBeta A will take (0, 1)

AlphaBeta B will take (0, 2)

AlphaBeta A will take (2, 0)

AlphaBeta B will take (2, 1)

AlphaBeta A will take (2, 2)

Game Over!

['A(1)', 'A(0)', 'B(1)']

['B(2)', 'A(4)', 'A(2)']

['A(1)', 'A(0)', 'A(1)']

Player A : 9 points, 2471 nodes in 0.5 total seconds and 5 moves.

Player B : 3 points, 1379 nodes in 0.3 total seconds and 4 moves.

Player A wins!

**Alphabeta-Minimax:**

$ python candyGame.py -b small.txt -p1 alphabeta -p2 minimax

AlphaBeta A will take (1, 1)

Minimax B will take (1, 0)

AlphaBeta A will take (1, 2)

Minimax B will take (0, 0)

AlphaBeta A will take (0, 1)

Minimax B will take (0, 2)

AlphaBeta A will take (2, 0)

Minimax B will take (2, 1)

AlphaBeta A will take (2, 2)

Game Over!

['A(1)', 'A(0)', 'B(1)']

['B(2)', 'A(4)', 'A(2)']

['A(1)', 'A(0)', 'A(1)']

Player A : 9 points, 2471 nodes in 0.5 total seconds and 5 moves.

Player B : 3 points, 600 nodes in 0.1 total seconds and 4 moves.

Player A wins!

**Minimax-Alphabeta:**

$ python candyGame.py -b small.txt -p1 minimax -p2 alphabeta

Minimax A will take (1, 1)

AlphaBeta B will take (1, 0)

Minimax A will take (1, 2)

AlphaBeta B will take (0, 0)

Minimax A will take (0, 1)

AlphaBeta B will take (0, 2)

Minimax A will take (2, 0)

AlphaBeta B will take (2, 1)

Minimax A will take (2, 2)

Game Over!

['A(1)', 'A(0)', 'B(1)']

['B(2)', 'A(4)', 'A(2)']

['A(1)', 'A(0)', 'A(1)']

Player A : 9 points, 945 nodes in 0.2 total seconds and 5 moves.

Player B : 3 points, 1379 nodes in 0.3 total seconds and 4 moves.

Player A wins!