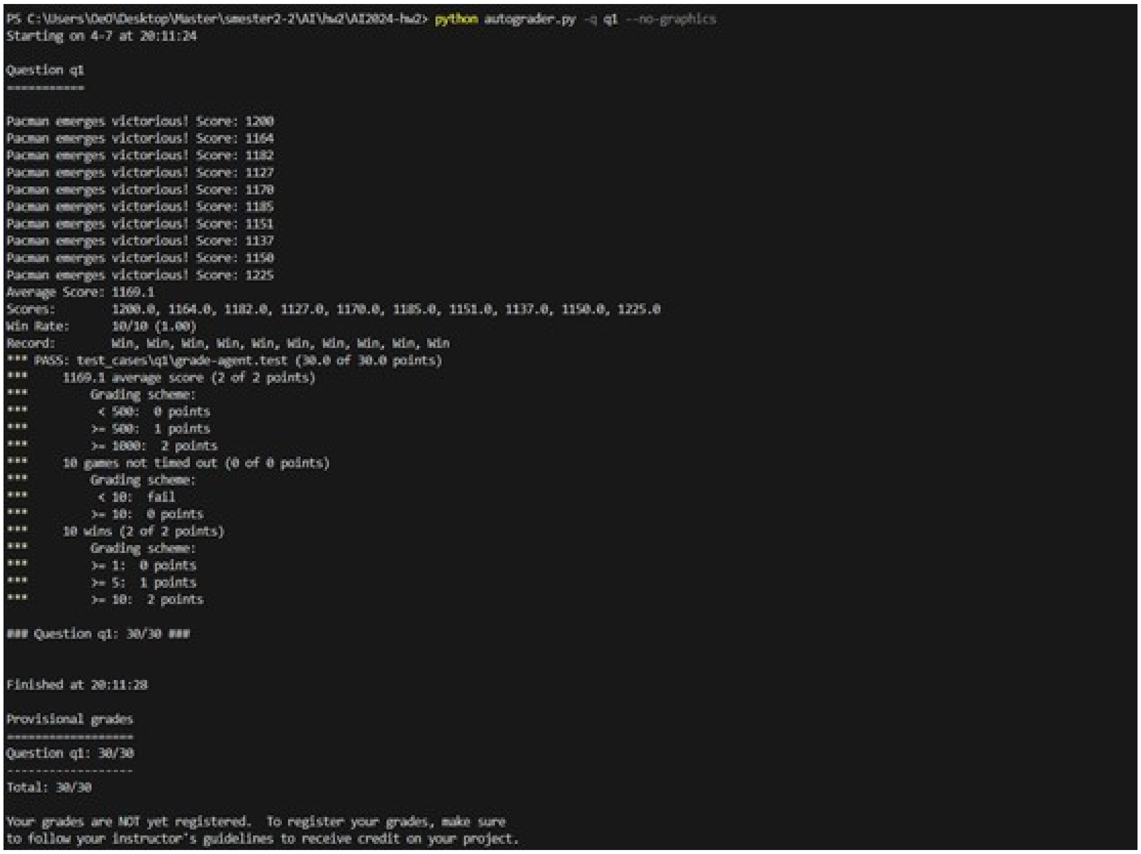
AI HW2

洪郡辰
R11944050

April 7, 2024

Q1

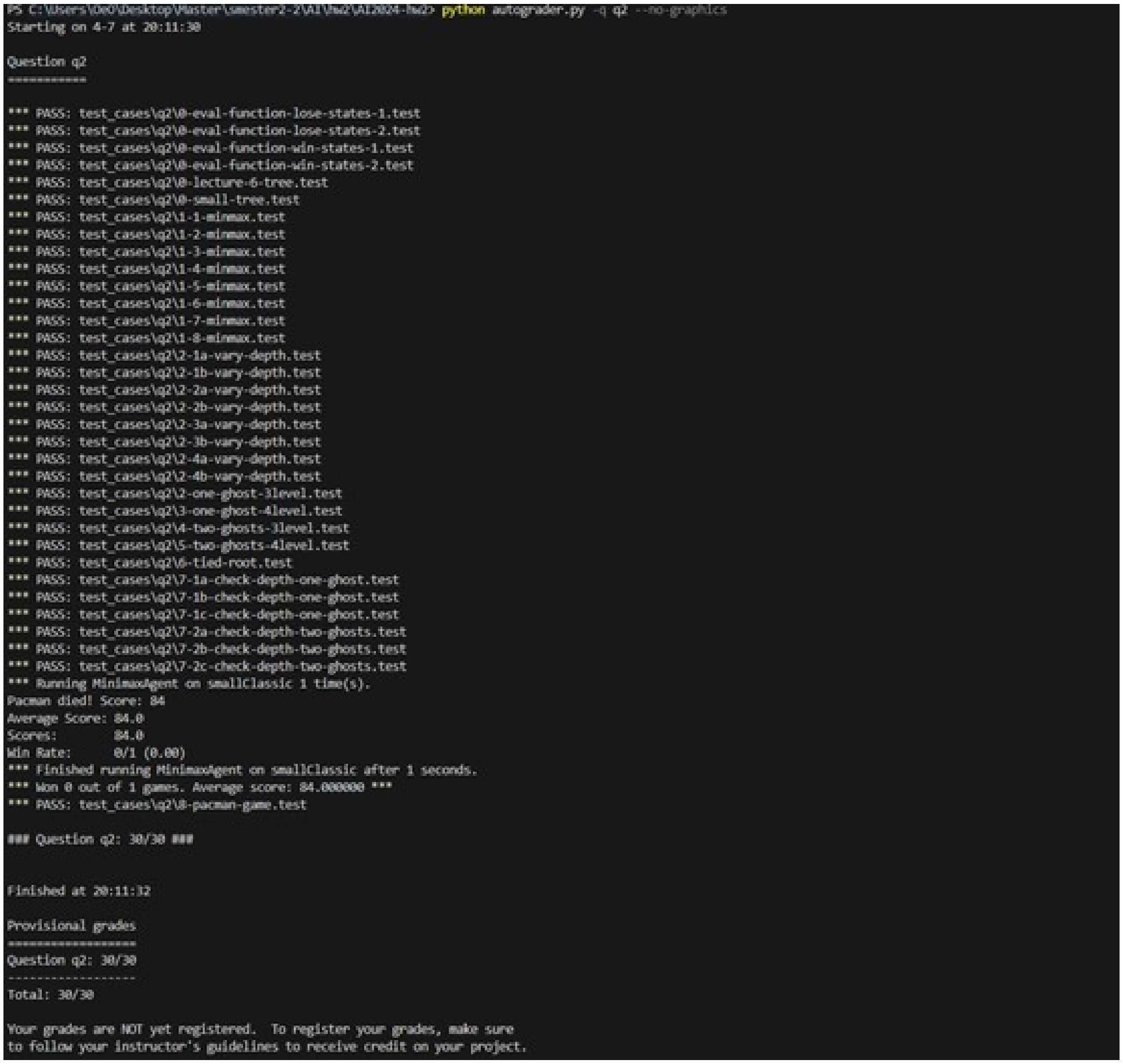


```
PS C:\Users\OeO\Desktop\Master\smester2-2\AI\hw2\AI2024-hw2> python autograder.py -q q1 --no-graphics
Starting on 4-7 at 20:11:24

Question q1
===========

Pacman emerges victorious! Score: 1200
Pacman emerges victorious! Score: 1164
Pacman emerges victorious! Score: 1182
Pacman emerges victorious! Score: 1127
Pacman emerges victorious! Score: 1170
Pacman emerges victorious! Score: 1185
Pacman emerges victorious! Score: 1151
Pacman emerges victorious! Score: 1137
Pacman emerges victorious! Score: 1150
Pacman emerges victorious! Score: 1225
Average Score: 1169.1
Scores:        1200.0, 1164.0, 1182.0, 1127.0, 1170.0, 1185.0, 1151.0, 1137.0, 1150.0, 1225.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases\q1\grade-agent.test (30.0 of 30.0 points)
***     1169.1 average score (2 of 2 points)
***         Grading scheme:
***          < 500:  0 points
***         >= 500:  1 points
***         >= 1000:  2 points
***     10 games not timed out (0 of 0 points)
***         Grading scheme:
***          < 10:  fail
***         >= 10:  0 points
***     10 wins (2 of 2 points)
***         Grading scheme:
***         >= 1:  0 points
***         >= 5:  1 points
***         >= 10:  2 points

### Question q1: 30/30 ###


Finished at 20:11:28

Provisional grades
==================
Question q1: 30/30
------------------
Total: 30/30

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

Reflex Agent 是基於 current state 去選擇 action 的一種策略，它會根據目前的 state 基於我設計的規則去選擇 action。首先我先排除掉 stop，不讓 pacman 執行 stop。第二點是保證 pacman 安全，如果當 pacman 與 ghost 的 manhattan distance 小於 3 時，pacman 優先選擇遠離 ghost 的 action，不選擇靠近 ghost 的 action。如果 pacman 處在無敵狀態則無視 ghost 位置。第三點是制定前往 food

位置的策略，因為優先選擇分數高的 action，我定義分數為 pacman 與最近的 food 的 manhattan distance 的倒數，我計算 current state 的分數以及執行 action 後的分數並透過機率的方式去選擇最終給定的值，並加入分數等於 1(最高優先) 在這三個分數中選擇，這三種分數被選擇到的機率不相同，選擇到的分數會定義 action 的分數。

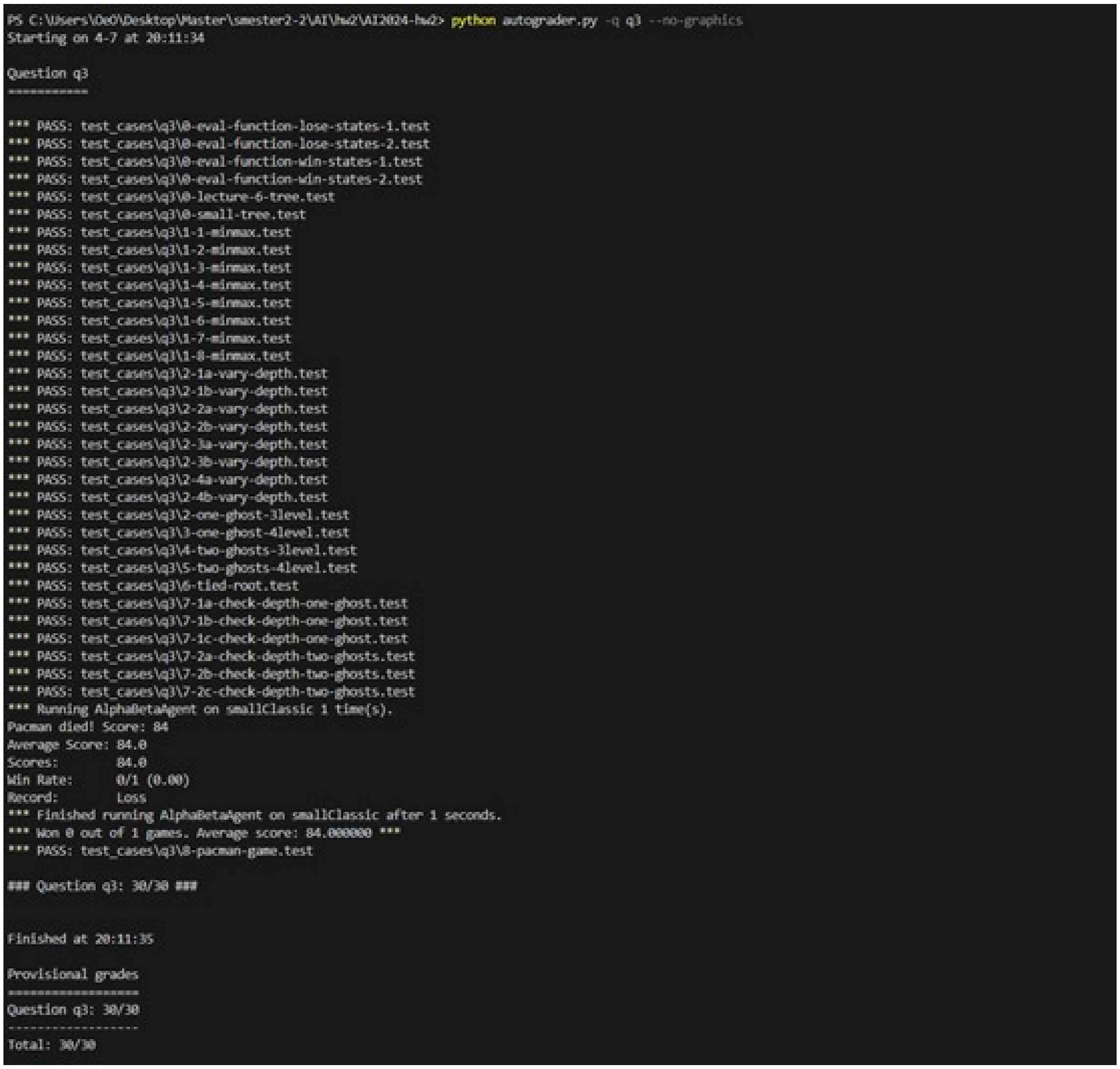
Q2



```
PS C:\Users\OeO\Desktop\Master\smester2-2\AI\hw2\AI2024-hw2> python autograder.py -q q2 --no-graphics
Starting on 4-7 at 20:11:30

Question q2
===========

*** PASS: test_cases\q2\0-eval-function-lose-states-1.test
*** PASS: test_cases\q2\0-eval-function-lose-states-2.test
*** PASS: test_cases\q2\0-eval-function-win-states-1.test
*** PASS: test_cases\q2\0-eval-function-win-states-2.test
*** PASS: test_cases\q2\0-lecture-6-tree.test
*** PASS: test_cases\q2\0-small-tree.test
*** PASS: test_cases\q2\1-1-minmax.test
*** PASS: test_cases\q2\1-2-minmax.test
*** PASS: test_cases\q2\1-3-minmax.test
*** PASS: test_cases\q2\1-4-minmax.test
*** PASS: test_cases\q2\1-5-minmax.test
*** PASS: test_cases\q2\1-6-minmax.test
*** PASS: test_cases\q2\1-7-minmax.test
*** PASS: test_cases\q2\1-8-minmax.test
*** PASS: test_cases\q2\2-1a-vary-depth.test
*** PASS: test_cases\q2\2-1b-vary-depth.test
*** PASS: test_cases\q2\2-2a-vary-depth.test
*** PASS: test_cases\q2\2-2b-vary-depth.test
*** PASS: test_cases\q2\2-3a-vary-depth.test
*** PASS: test_cases\q2\2-3b-vary-depth.test
*** PASS: test_cases\q2\2-4a-vary-depth.test
*** PASS: test_cases\q2\2-4b-vary-depth.test
*** PASS: test_cases\q2\2-one-ghost-3level.test
*** PASS: test_cases\q2\3-one-ghost-4level.test
*** PASS: test_cases\q2\4-two-ghosts-3level.test
*** PASS: test_cases\q2\5-two-ghosts-4level.test
*** PASS: test_cases\q2\6-tied-root.test
*** PASS: test_cases\q2\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1b-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2c-check-depth-two-ghosts.test
*** Running MinimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:        84.0
Win Rate:      0/1 (0.00)
*** Finished running MinimaxAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases\q2\8-pacman-game.test

### Question q2: 30/30 ###


Finished at 20:11:32

Provisional grades
==================
Question q2: 30/30
------------------
Total: 30/30

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

Minimax 是一種觀察連續執行多次 action 後選擇目前 action 的一種策略，它會定義 depth，depth 代表連續觀察幾次 action，推斷出 depth 內所有可能發生的狀態，找到最佳策略，使自己的利益最大化同時對手的利益最小化。在 minimax 中會對每個可能發生的狀態計算出一個分數，這個分數代表在該選擇下遊戲的結果，MAX 玩家 (自己) 會選擇分數最大的選擇，MIN 玩家 (對手) 會選擇分數最小的選擇。

Q3



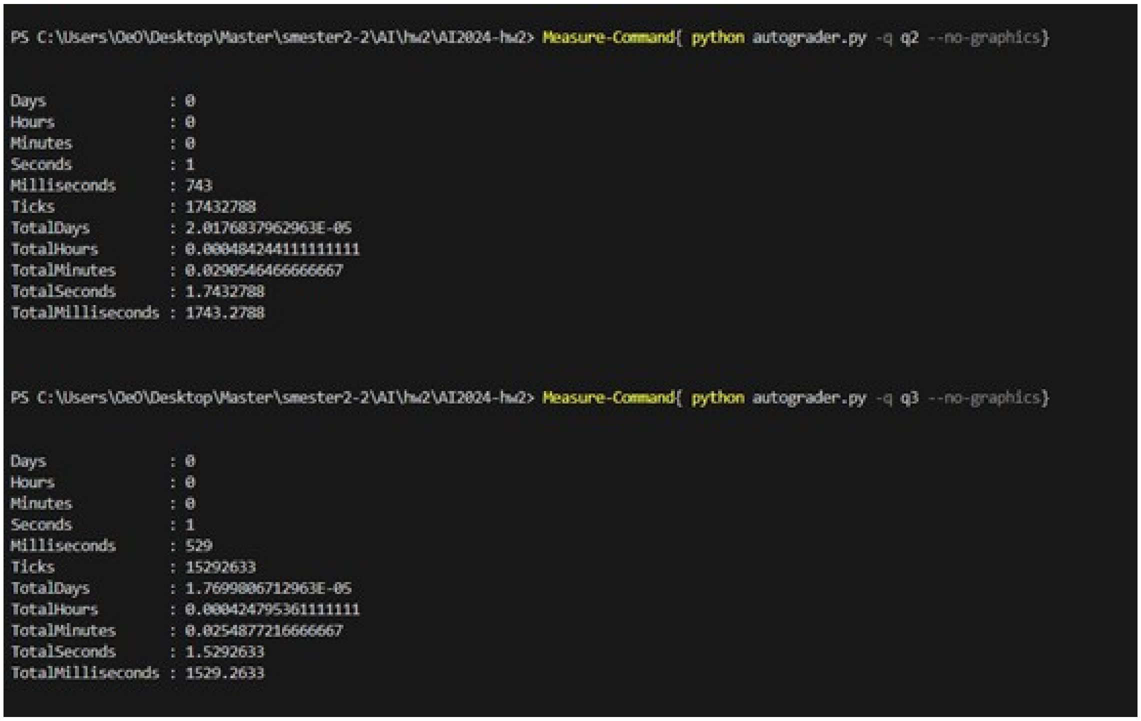
```
PS C:\Users\OeO\Desktop\Master\smester2-2\AI\hw2\AI2024-hw2> python autograder.py -q q3 --no-graphics
Starting on 4-7 at 20:11:34

Question q3
===========

*** PASS: test_cases\q3\0-eval-function-lose-states-1.test
*** PASS: test_cases\q3\0-eval-function-lose-states-2.test
*** PASS: test_cases\q3\0-eval-function-win-states-1.test
*** PASS: test_cases\q3\0-eval-function-win-states-2.test
*** PASS: test_cases\q3\0-lecture-6-tree.test
*** PASS: test_cases\q3\0-small-tree.test
*** PASS: test_cases\q3\1-1-minmax.test
*** PASS: test_cases\q3\1-2-minmax.test
*** PASS: test_cases\q3\1-3-minmax.test
*** PASS: test_cases\q3\1-4-minmax.test
*** PASS: test_cases\q3\1-5-minmax.test
*** PASS: test_cases\q3\1-6-minmax.test
*** PASS: test_cases\q3\1-7-minmax.test
*** PASS: test_cases\q3\1-8-minmax.test
*** PASS: test_cases\q3\2-1a-vary-depth.test
*** PASS: test_cases\q3\2-1b-vary-depth.test
*** PASS: test_cases\q3\2-2a-vary-depth.test
*** PASS: test_cases\q3\2-2b-vary-depth.test
*** PASS: test_cases\q3\2-3a-vary-depth.test
*** PASS: test_cases\q3\2-3b-vary-depth.test
*** PASS: test_cases\q3\2-4a-vary-depth.test
*** PASS: test_cases\q3\2-4b-vary-depth.test
*** PASS: test_cases\q3\2-one-ghost-3level.test
*** PASS: test_cases\q3\3-one-ghost-4level.test
*** PASS: test_cases\q3\4-two-ghosts-3level.test
*** PASS: test_cases\q3\5-two-ghosts-4level.test
*** PASS: test_cases\q3\6-tied-root.test
*** PASS: test_cases\q3\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-1b-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q3\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q3\7-2c-check-depth-two-ghosts.test
*** Running AlphaBetaAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:        84.0
Win Rate:      0/1 (0.00)
Record:        Loss
*** Finished running AlphaBetaAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases\q3\8-pacman-game.test

### Question q3: 30/30 ###


Finished at 20:11:35

Provisional grades
==================
Question q3: 30/30
------------------
Total: 30/30
```

Alpha-beta pruning 是一種用於提升 Minimax 演算法效率的技術，在 minimax 中會遍歷所有可能的選擇，這在搜索空間很大時會變得非常耗時，Alpha-beta pruning 通過刪除一些無關緊要的節點，提升 minimax 演算法的效率。alpha-beta pruning 通過在 minimax 中加入 alpha 及 beta 分別代表 MAX 玩家已知最佳解及 MIN 玩家已知最佳解。在遍歷決策樹時，當發現某個節點的值超出父節點的 alpha 及 beta 區間時即可進行 pruning，減少需要搜索的節點數量。

Q4

首先我先排除掉 stop，不讓 pacman 執行 stop。第二點是保證 pacman 安全，如果當 pacman 與 ghost 的 manhattan distance 小於 3 時，pacman 優先選擇遠離 ghost 的 action，不選擇靠近 ghost 的 action。如果 pacman 處在無敵狀態則無視 ghost 位置。第三點是制定前往 food 位置的策略，因為優先選擇分數高的 action，我定義分數為 pacman 與最近的 food 的 manhattan distance 的倒數，我計算 current state 的分數以及執行 action 後的分數並透過機率的方式去選擇最終給定的值，並加入分數等於 1(最高優先) 在這三個分數中選擇，這三種分數被選擇到的機率不相同，選擇到的分數會定義 action 的分數。

Q5



```
PS C:\Users\OeO\Desktop\Master\smester2-2\AI\hw2\AI2024-hw2> Measure-Command{ python autograder.py -q q2 --no-graphics}


Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 1
Milliseconds      : 743
Ticks             : 17432788
TotalDays         : 2.0176837962963E-05
TotalHours        : 0.000484244111111111
TotalMinutes      : 0.0290546466666667
TotalSeconds      : 1.7432788
TotalMilliseconds : 1743.2788



PS C:\Users\OeO\Desktop\Master\smester2-2\AI\hw2\AI2024-hw2> Measure-Command{ python autograder.py -q q3 --no-graphics}


Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 1
Milliseconds      : 529
Ticks             : 15292633
TotalDays         : 1.7699806712963E-05
TotalHours        : 0.000424795361111111
TotalMinutes      : 0.0254877216666667
TotalSeconds      : 1.5292633
TotalMilliseconds : 1529.2633
```

我透過 Measure-Command 計算執行 autograder q2 及 q3 的 runtime，可以看到在相同問題下 minimax 的 runtime 大於 alpha-beta pruning 後的 runtime。