TCG homework 2 report

學生：莊博任

學號：R05922107

使用語言：C++

compile方式：makefile

實作內容：

1. UCT tree: struct Node{  
    int Board[BOUNDARYSIZE][BOUNDARYSIZE]; //node 盤面

Node \*child[HISTORYLENGTH];

Node \*parent;

double ucb\_score;

double win;

double sim\_time; //該node simulation次數

int num\_legal\_moves; //下一手總合法走步數

int num\_after\_prunned; //pruned過後的合法走步數

int turn; //下一手輪誰下

int GameRecord[MAXGAMELENGTH][BOUNDARYSIZE][BOUNDARYSIZE]; // 歷史盤面紀錄

int game\_length; //雙方共走了幾手

int move; //此盤面上一手下哪（主要用於debug）

bool expended; //是否已經被expend過

bool prunned; //是否被pruned，1則代表此node不該再被當作合法盤面  
}

1. MCTS algorithm function:   
   select()  
   expend()  
   simulate() //每一手simulate，先用隨機位置找30次，都失敗才用gen\_legal\_move() (也有優化檢查歷史盤面)  
   back\_propagation()  
   sim\_all\_children() //simulate select node的所有child  
   init\_root() //用於產生每步的root  
   print\_pv() //紀錄pv path 於 PV\_path.txt (只記錄最終tree的select過程，用ucb score判斷，可能結果跟實際下的結果不同（實際下用win rate判斷）)  
   mcts\_ucb\_pick\_move() //將上述function 包起來並判斷最終落子位置，還有決定落子位置之後delete tree。
2. Progressive pruning function:  
   prunning() //實作結果不好
3. 優化原template function:  
   count\_liberty() //改 total\_liberty > 1即回傳  
   record() //使用std::copy複製record, 並記錄盤面總共子數、總黑子數、總白子數  
   gen\_legal\_move() //檢查重複歷史盤面時，先檢查總子數是否相同，再檢查黑子數是否相同，都相同才用原本的線性檢查法。  
   gen\_move() //將原本的記錄時間改用chrono::system\_clock::time\_point，以記錄真實時間，以確保在9秒內做完。

最終成果：  
我執黑跟我做的AI下，輸9.5目……  
  
心得：

幾乎是第一次認真寫C++ code，以前都是寫JAVA code，查了很多c++的function和記憶體相關用法。花了很多時間在debug select和simulate的turn問題，一直搞錯勝率是己方還是對方的。優化的問題問了助教之後改了count\_liberty和simulate先用隨機選取落子位置，但是查了很久都沒辦法實作出用set檢查歷史盤面，後來想到先檢查總子數的方法，總算把10秒內simulate 三千次變成兩萬次左右。