300103 Data Structures and Algorithms

Practical Exercise 4

(no marks, due in Week 5)

Task 4.1
In Practical 3 Task 3.2, we were required to implement a function, named `pushDown()`, for the class State in order to to push down all the non-zero numbers of each column towards the bottom. I implement it as the following:

```
//Wrong solution for Task 3.2
void State::pushdown_Wrong() {

      for (int j = 0; j < BOARDSIZE; j++) {
            stack<int> tempStack;
            for (int i = 0; i < BOARDSIZE; i++) {
                  if (grid[i][j] != 0)
                        tempStack.push(grid[i][j]);
            }

            for (int i = 0; i < BOARDSIZE; i++) {
                  if (!tempStack.empty()) {
                        grid[i][j] = tempStack.top();
                        tempStack.pop();
                  } else {
                        grid[i][j] = 0;
                  }
            }
      }
}
```

The function can push the numbers down but unfortunately the ordering of values in each column is reversed. Help me to fix the problem by using a *queue* instead of a *stack*.

Task 4.2
Download *TicTacToe.zip* from Code for Practical 4. Read the definition for struct Cell. Explain to your tutor the functionality of *operator<*:

```
struct Cell {
   int x;
   int y;
   int heuristic;

   Cell(int xx, int yy, int hh):x(xx), y(yy), heuristic(hh) {}
```

```
    bool operator<(const Cell& c) const {
      return heuristic < c.heuristic;
    }
};
```

Task 4.3
Read the definition of class BestFirstPlayer in file BestFirstPlayer.h. Convert the definition of function getMove into pseudo-code and explain the algorithm to your tutor. Assume that the size of board is not 3 but is *n*. What is the worst-case complexity of this algorithm?

Hint: *Assume that the complexity for push and pop operation in priority queue is O(logn), respectively while the completion of top operation is O(1).*

Task 4.4
Improve the implementation of heuristics function in class TicTacToe to make the BestFirstPlayer smarter. Compare your BestFirstPlayer with my BestFristPlayer to see if your implementation has higher intelligence.