Kumar Satyam

21052985

**ADVERSARIAL SEARCH**

#include <stdio.h>

#include <string.h>

#define N 4

int utility(char \*board) {

int count = 0;

for (int i = 0; i < N; i++) {

if (board[i] == 'B') {

count++;

}

}

return count;

}

int is\_terminal(char \*board) {

for (int i = 0; i < N; i++) {

if (board[i] == '0') {

return 0; // Game not in terminal state

}

}

return 1; // Game in terminal state

}

int max\_value(char \*board);

int min\_value(char \*board, int action) {

char new\_board[N];

memcpy(new\_board, board, N);

new\_board[action] = 'W'; // Player 2 (MIN) places a white coin

// Flip adjacent coins

if (action > 0 && new\_board[action - 1] == 'B') {

new\_board[action - 1] = 'W';

}

if (action < N - 1 && new\_board[action + 1] == 'B') {

new\_board[action + 1] = 'W';

}

if (is\_terminal(new\_board)) {

return utility(new\_board);

}

int min\_val = N + 1;

for (int i = 0; i < N; i++) {

if (new\_board[i] == '0') {

int max\_val = max\_value(new\_board);

if (max\_val < min\_val) {

min\_val = max\_val;

}

}

}

return min\_val;

}

int max\_value(char \*board) {

char new\_board[N];

memcpy(new\_board, board, N);

int max\_val = -1;

for (int i = 0; i < N; i++) {

if (new\_board[i] == '0') {

new\_board[i] = 'B'; // Player 1 (MAX) places a black coin

// Flip adjacent coins

if (i > 0 && new\_board[i - 1] == 'W') {

new\_board[i - 1] = 'B';

}

if (i < N - 1 && new\_board[i + 1] == 'W') {

new\_board[i + 1] = 'B';

}

if (is\_terminal(new\_board)) {

int utility\_val = utility(new\_board);

if (utility\_val > max\_val) {

max\_val = utility\_val;

}

} else {

for (int j = 0; j < N; j++) {

if (new\_board[j] == '0') {

int min\_val = min\_value(new\_board, j);

if (min\_val > max\_val) {

max\_val = min\_val;

}

}

}

}

}

}

return max\_val;

}

int main() {

char board[] = "0000W00BBW0";

int max\_utility = max\_value(board);

printf("Utility of Player 1 (MAX): %d\n", max\_utility);

return 0;

}