**ADA LAB WEEK 8**

**Yash Gupta**

**1BM21CS251**

**22/08/23**

**1) N Queens Problem using C**

#include <stdio.h>

#include <stdlib.h>

void displayBoard(char board[][10], int n) {

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

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

printf("%c ", board[i][j]);

}

printf("\n");

}

}

int isSafe(int row, int col, char board[][10], int n) {

int duprow = row;

int dupcol = col;

while (col >= 0) {

if (board[row][col] == 'Q')

return 0;

col--;

}

row = duprow;

col = dupcol;

while (row >= 0 && col >= 0) {

if (board[row][col] == 'Q')

return 0;

row--;

col--;

}

row = duprow;

col = dupcol;

while (row < n && col >= 0) {

if (board[row][col] == 'Q')

return 0;

row++;

col--;

}

return 1;

}

void solve(int col, char board[][10], int n) {

if (col == n) {

displayBoard(board, n);

printf("\n"); // For next combination of board

return;

}

for (int row = 0; row < n; row++) {

if (isSafe(row, col, board, n)) {

board[row][col] = 'Q';

solve(col + 1, board, n);

board[row][col] = '.'; // Backtracking step

}

}

}

int main() {

int n;

printf("Enter the dimension of chessBoard\n");

scanf("%d", &n);

if (n < 4 && n!=1) {

printf("No solution exists\n");

exit(0);

}

char board[10][10];

// Initialising board with No queen

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

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

board[i][j] = '.';

}

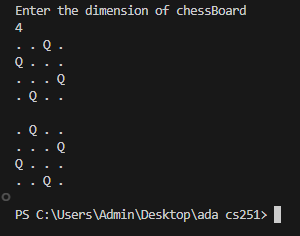
}

solve(0, board, n); // 0th col is called

return 0;

}

**Output:**

****

**2) Heap Sorting Technique using C**

#include <stdio.h>

#include <time.h>

void swap(int \*a, int \*b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void heapify(int a[], int n, int i)

{

int largest = i;

int l = 2 \* i + 1;

int r = 2 \* i + 2;

if (l < n && a[l] > a[largest])

{

largest = l;

}

if (r < n && a[r] > a[largest])

{

largest = r;

}

if (largest != i)

{

swap(&a[i], &a[largest]);

heapify(a, n, largest);

}

}

void heapSort(int a[], int n)

{

for (int i = n / 2 - 1; i >= 0; i--)

{

heapify(a, n, i);

}

for (int i = n - 1; i > 0; i--)

{

swap(&a[0], &a[i]);

heapify(a, i, 0);

}

}

int main()

{

int a[10001];

printf("Enter the size of array\n");

int n;

scanf("%d", &n);

printf("enter the array elements:\n");

for (int i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

printf("Original array: ");

for (int i = 0; i < n; i++)

{

printf("%d ", a[i]);

}

printf("\n");

heapSort(a, n);

printf("Sorted array: ");

for (int i = 0; i < n; i++)

{

printf("%d ", a[i]);

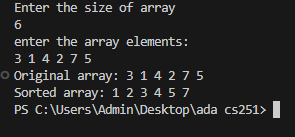
}

printf("\n");

return 0;

}

**Output:**

****