Lab Assignment 3

Course Title: Algorithm Design and Analysis Lab Work

Course Code: CSE-2264

Submitted to:

MD. Mahadi Hasan

Lecturer, Department of CSE

Northern University Bangladesh

Submitted by:

Name: Sadia Islam

ID: 41220300363

Section: 4B

Program: CSE

Department of Computer Science Engineering

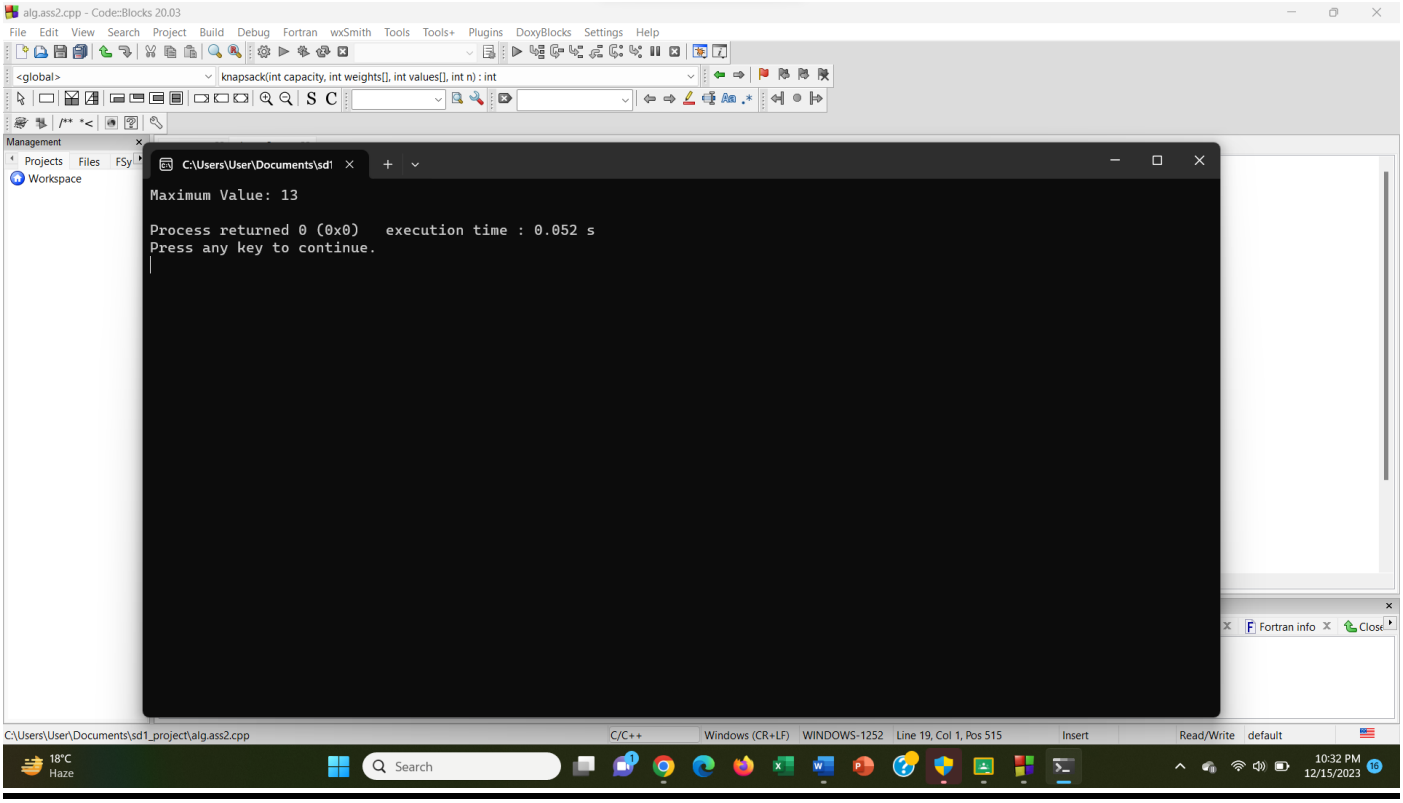
Date of submission: 15 December 2023

NORTHERN UNIVERSITY BANGLADESH

Problem 1:

C++ code:

```
#include <iostream>
using namespace std;

int knapsack(int capacity, int weights[], int values[], int n) {
    int dp[n + 1][capacity + 1];



    for (int i = 0; i <= n; i++) {
        for (int w = 0; w <= capacity; w++) {
            if (i == 0 || w == 0)
                dp[i][w] = 0;
            else if (weights[i - 1] <= w)
                dp[i][w] = max(dp[i - 1][w], values[i - 1] + dp[i - 1][w - weights[i - 1]]);
            else
                dp[i][w] = dp[i - 1][w];
        }
    }
   return dp[n][capacity];
}

int main() {
    int knapsack_capacity = 10;
    int item_weights[] = {2, 3, 4, 5};
    int item_values[] = {3, 4, 5, 6};
    int n = sizeof(item_weights) / sizeof(item_weights[0]);

    int result = knapsack(knapsack_capacity, item_weights, item_values, n);

    cout << "Maximum Value: " << result << endl;

    return 0;
}
```

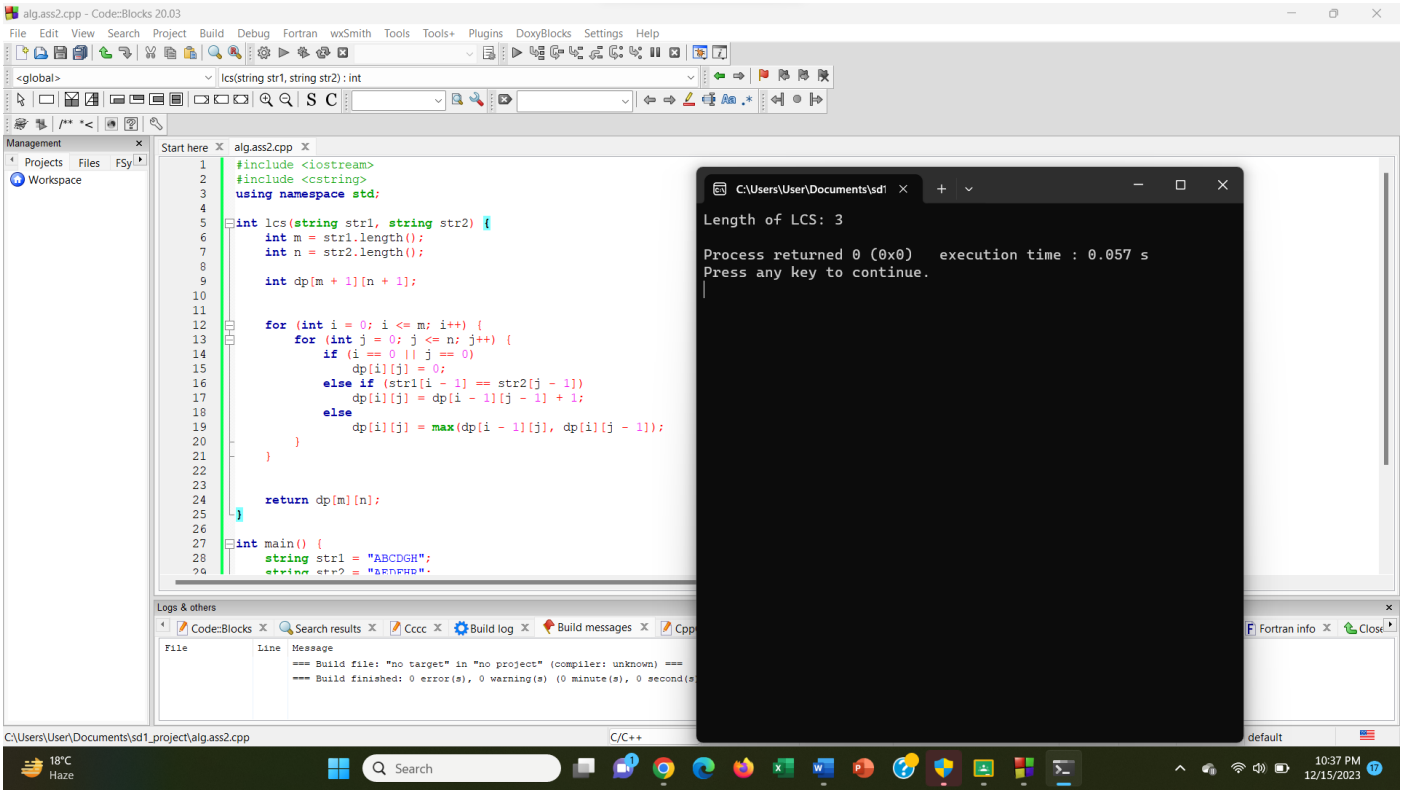


Problem 2:

C++ Code:

```
#include <iostream>
#include <cstring>
using namespace std;

int lcs(string str1, string str2) {
    int m = str1.length();
    int n = str2.length();

    int dp[m + 1][n + 1];


    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (str1[i - 1] == str2[j - 1])
                dp[i][j] = dp[i - 1][j - 1] + 1;
            else
```

```
            dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }


    return dp[m][n];
}

int main() {
    string str1 = "ABCDGH";
    string str2 = "AEDFHR";

    int result = lcs(str1, str2);

    cout << "Length of LCS: " << result << endl;

    return 0;
}
```

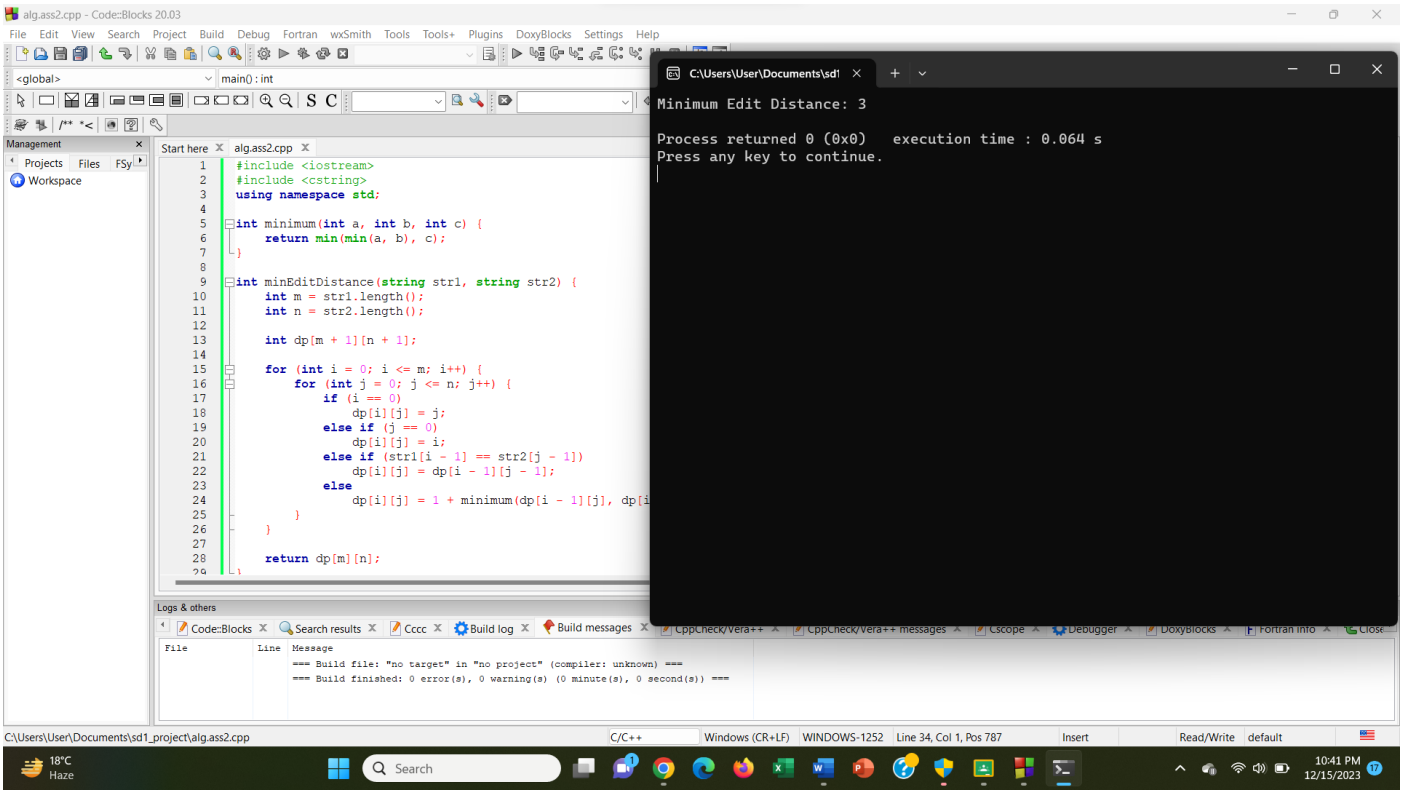


Problem 3:

C++ Code:

```
#include <iostream>
#include <cstring>
using namespace std;

int minimum(int a, int b, int c) {
    return min(min(a, b), c);
}

int minEditDistance(string str1, string str2) {
    int m = str1.length();
    int n = str2.length();

    int dp[m + 1][n + 1];

    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0)
                dp[i][j] = j;
            else if (j == 0)
                dp[i][j] = i;
            else if (str1[i - 1] == str2[j - 1])
                dp[i][j] = dp[i - 1][j - 1];
            else
                dp[i][j] = 1 + minimum(dp[i - 1][j], dp[i][j - 1], dp[i - 1][j - 1]);
        }
    }

    return dp[m][n];
}

int main() {
    string str1 = "kitten";
    string str2 = "sitting";
```

```
    int result = minEditDistance(str1, str2);

    cout << "Minimum Edit Distance: " << result << endl;

    return 0;
}
```



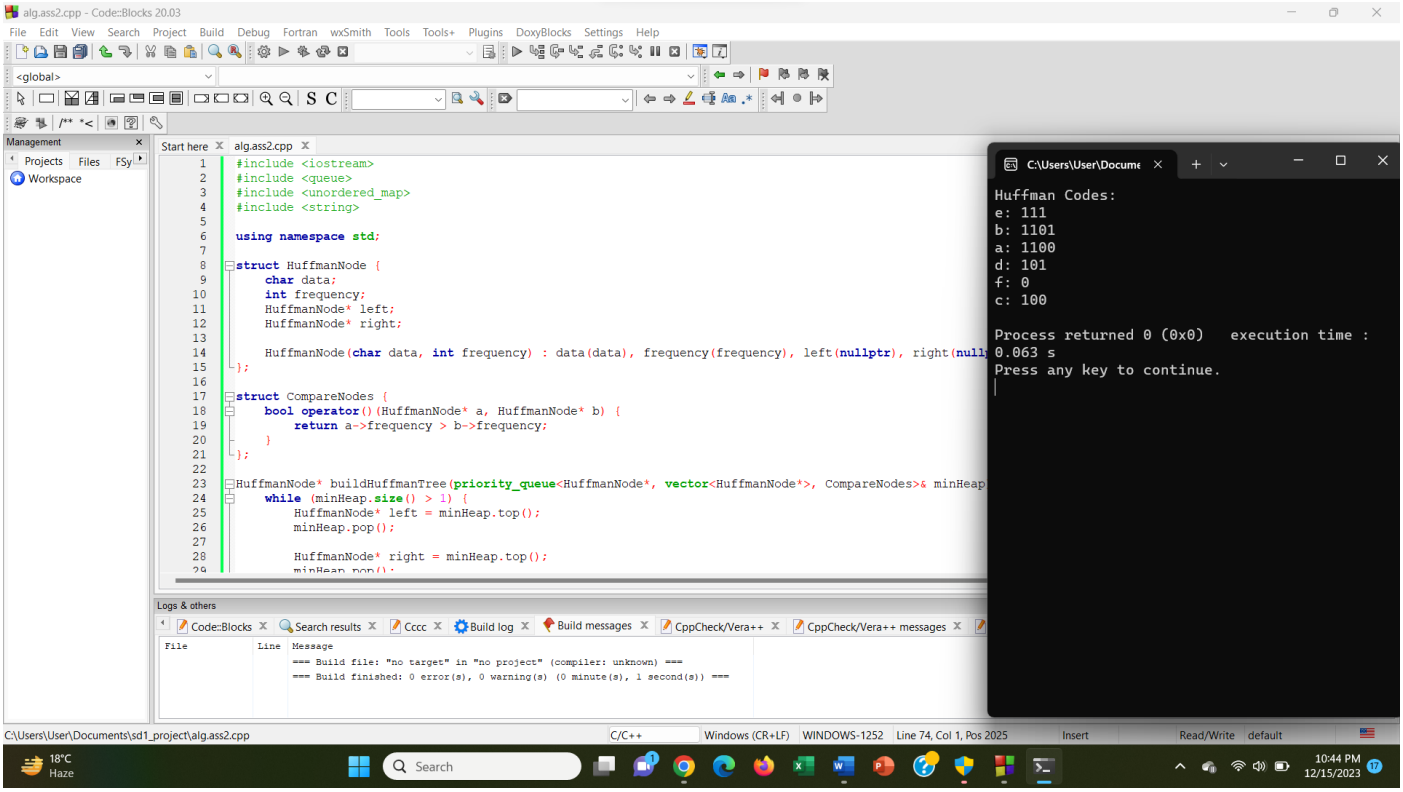
Problem 4:

C++ Code:

```
#include <iostream>
#include <queue>
#include <unordered_map>
#include <string>

using namespace std;

struct HuffmanNode {
    char data;
    int frequency;
```

```
    HuffmanNode* left;
    HuffmanNode* right;

    HuffmanNode(char data, int frequency) : data(data), frequency(frequency), left(nullptr), right(nullptr) {}
};

struct CompareNodes {
    bool operator()(HuffmanNode* a, HuffmanNode* b) {
        return a->frequency > b->frequency;
    }
};

HuffmanNode* buildHuffmanTree(priority_queue<HuffmanNode*, vector<HuffmanNode*>,
CompareNodes>& minHeap) {
    while (minHeap.size() > 1) {
        HuffmanNode* left = minHeap.top();
        minHeap.pop();

        HuffmanNode* right = minHeap.top();
        minHeap.pop();

        HuffmanNode* internalNode = new HuffmanNode('$', left->frequency + right->frequency);
        internalNode->left = left;
        internalNode->right = right;

        minHeap.push(internalNode);
    }

    return minHeap.top();
}

void generateHuffmanCodes(HuffmanNode* root, string code, unordered_map<char, string>&
huffmanCodes) {
    if (!root)
        return;
```

```
    if (root->data != '$') {
        huffmanCodes[root->data] = code;
    }

    generateHuffmanCodes(root->left, code + "0", huffmanCodes);
    generateHuffmanCodes(root->right, code + "1", huffmanCodes);
}

int main() {
    unordered_map<char, int> charFrequencies = {{'a', 5}, {'b', 9}, {'c', 12}, {'d', 13}, {'e', 16}, {'f', 45}};

    priority_queue<HuffmanNode*, vector<HuffmanNode*>, CompareNodes> minHeap;

    for (const auto& entry : charFrequencies) {
        minHeap.push(new HuffmanNode(entry.first, entry.second));
    }

    HuffmanNode* root = buildHuffmanTree(minHeap);

    unordered_map<char, string> huffmanCodes;
    generateHuffmanCodes(root, "", huffmanCodes);

    cout << "Huffman Codes:" << endl;
    for (const auto& entry : huffmanCodes) {
        cout << entry.first << ": " << entry.second << endl;
    }

    return 0;
}
```


alg.ass2.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSy
Workspace
Start here
alg.ass2.cpp
#include <iostream>
#include <queue>
#include <unordered_map>
#include <string>
using namespace std;
struct HuffmanNode {
char data;
int frequency;
HuffmanNode* left;
HuffmanNode* right;
HuffmanNode(char data, int frequency) : data(data), frequency(frequency), left(nullptr), right(null
};
struct CompareNodes {
bool operator()(HuffmanNode* a, HuffmanNode* b) {
return a->frequency > b->frequency;
}
};
HuffmanNode* buildHuffmanTree(priority_queue<HuffmanNode*, vector<HuffmanNode*>, CompareNodes>& minHeap
while (minHeap.size() > 1) {
HuffmanNode* left = minHeap.top();
minHeap.pop();
HuffmanNode* right = minHeap.top();
C:\Users\User\Docume
Huffman Codes:
e: 111
b: 1101
a: 1100
d: 101
f: 0
c: 100
Process returned 0 (0x0) execution time :
0.063 s
Press any key to continue.
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
File
Line
Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 1 second(s)) ===
C:\Users\User\Documents\sd1_project\alg.ass2.cpp
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 74, Col 1, Pos 2025
Insert
Read/Write
default
18°C
Haze
Search
10:44 PM
12/15/2023