



Assignment Topic: A^* Search Algorithm

Course Title: Artificial Intelligence

Course Code: SWE 323

Date of Submission: 25th March 2023

Submitted to:

Sayma Sultana Chowdhury

Assistant Professor, IICT, SUST

Submitted by:

Sumonta Saha Mridul

Reg No: 2019831056

Ans:-

Reg:- 2019831056

Given that,

State	$h(n)$	State	$h(n)$
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	386
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu	193
Fagaras	176	Vilcea	80
Giurgiu	77	Sibiu	253
Hirsova	151	Timisoara	329
Iasi	226	Unziceni	80
Lugoj	244	Valui	199
		Zerind	374

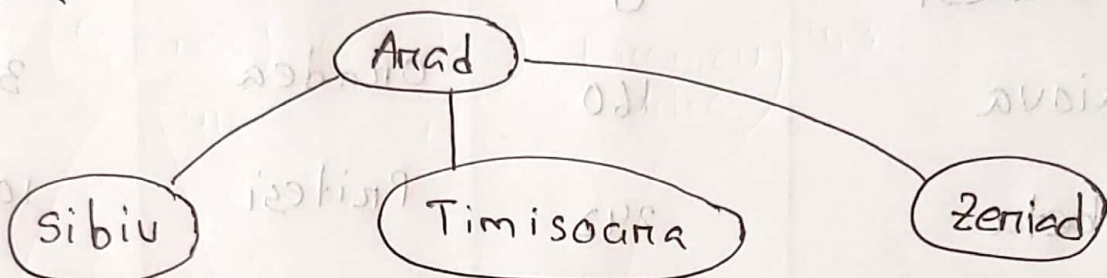
Steps to go Bucharest from Arad:-

Step-1:-

Arad

$$0 + 366 = 366$$

Step-2:- From expanding Arad,



$$140 + 253 = \boxed{393}$$

minimum

$$118 + 329 = 447$$

$$75 + 374 = 449$$

Step-3:- now expanding sibiu



$$140 + 140 + 366 = 646$$

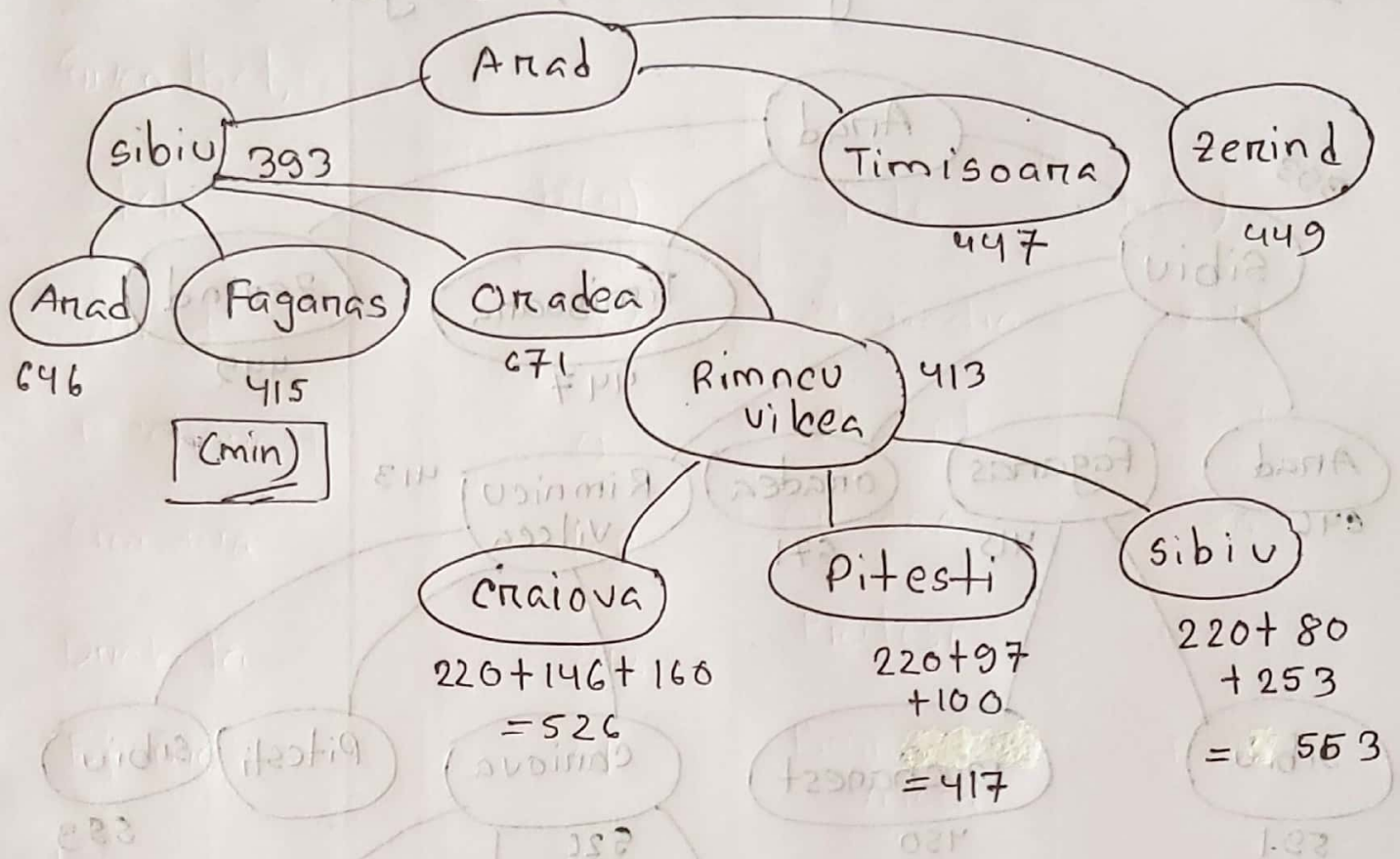
$$140 + 99 + 176 = 415$$

$$140 + 151 + 280 = 671$$

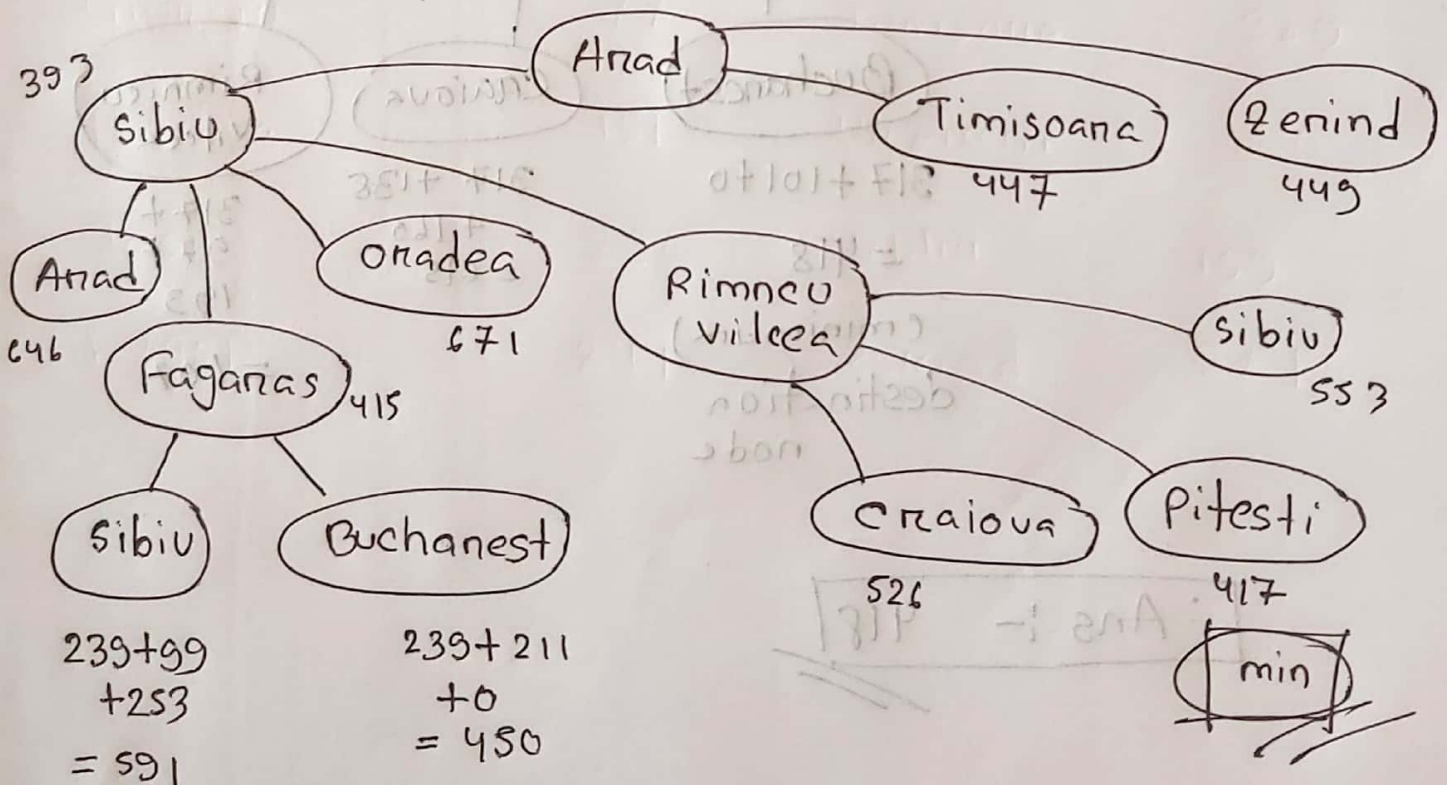
$$140 + 80 + 193 = \boxed{413}$$

minimum

step-4:- Expanding Rimnicu vilcea, we get

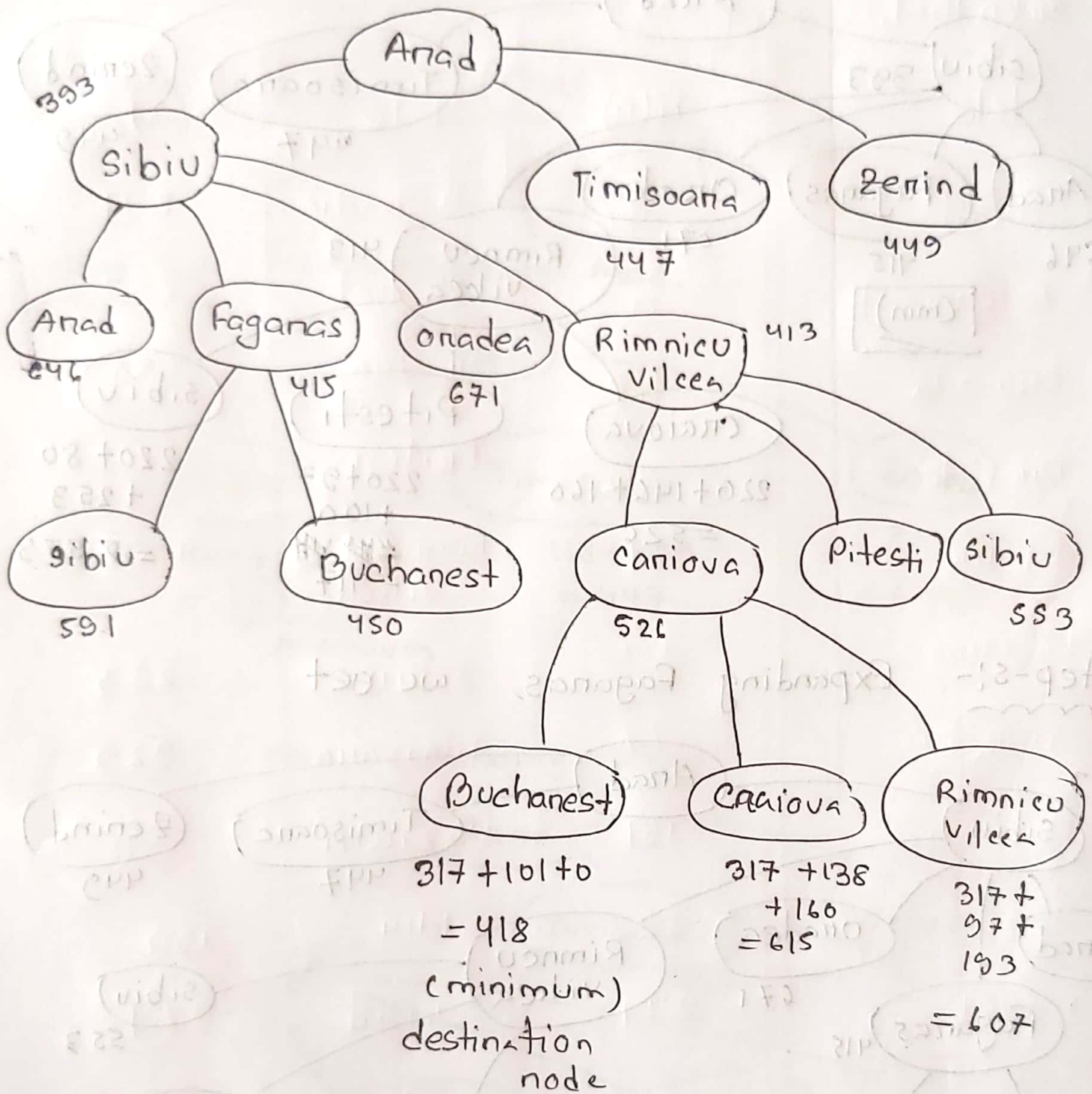


step-5:- Expanding Faganas, we get



Step - 6

Expanding Pitesti, we get



Ans :- 418

```

//      ** Sumonta Saha Mridul **                               SWE - SUST
/*

*      #####      ##      ##      ##      ##      #####      ##      ##      #####      ###
!      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
?      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
*      #####      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
!      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
?      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
*      #####      #####      ##      ##      #####      ##      ##      ##      ##      ##

*/

#include <bits/stdc++.h>
using namespace std;

#define ll long long
#define ull unsigned long long

#define f first
#define se second
#define lb lower_bound
#define ub upper_bound
#define pb push_back

#define pii pair<int, int>
#define pll pair<long, long>
#define mp(a, b) make_pair(a, b)
#define vi vector<int>
#define vll vector<ll>
#define vii vector<pii>
#define sll set<ll>
#define qll queue<ll>
#define stll stack<ll>
#define Mi map<int, int>
#define mii map<pii, int>
#define alls(a) (a).begin(), (a).end()
#define sz(x) (int)x.size()

#define Size(s) s.length()
#define all(v) (v.begin(), v.end)
#define rev(v) reverse(v.begin(), v.end)
#define srt(v) sort(v.begin(), v.end)

#define F(i, s, e) for (ll i = s; i < e; ++i)
#define rep(i, a, b) for (int i = a; i < b; i++)
#define rem(i, a, b) for (int i = a; i > b; i--)
#define pv(v) \
    for (ll i : v) \
        cout << i << " "; \
    cout << endl;

#define max3(a, b, c) max(max((a), (b)), (c))
#define min3(a, b, c) min(min((a), (b)), (c))

#define cin(n) cin >> n
#define cin2(a, b) cin >> a >> b;
#define cin3(a, b, c) cin >> a >> b >> c;
#define printE(str) cout << str << endl
#define print(str) cout << str << endl
#define endl '\n'
#define Y cout << "YES\n"

```

```

#define No cout << "NO\n"
#define ye cout << "Yes\n"
#define no cout << "No\n"

#define sp " "
#define enter cout << endl;

#define fast \
    ios_base::sync_with_stdio(false); \
    cin.tie(NULL); \
    cout.tie(NULL)

#define foi(i, a, b) for (int i = a; i < b; i++)
#define foI(i, a, b) for (int i = a; i <= b; i++)
#define fol(i, a, b) for (ll i = a; i < b; i++)
#define foL(i, a, b) for (ll i = a; i <= b; i++)

/* char a = 'A';   int num = (int) a;
   char a = '2';   int num = a-48;
   char a = '8';   int num = a - '0' ; ( s[i]-'0' == 8 )

const int MAXN = 100005;
const int INF = 1000000007;
int n, m;
vector<pair<int, int>> adj[MAXN];
int dist[MAXN];
int heuristic[MAXN];
int s, e;

void dijkstra(int start)
{
    priority_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> pq; // min-heap
    for (int i = 1; i <= n; i++)
        dist[i] = INF;
    dist[start] = 0;
    pq.push({0, start});
    while (!pq.empty())
    {
        int u = pq.top().second;
        pq.pop();
        for (auto v : adj[u])
        {
            int new_dist = dist[u] + v.second;
            if (new_dist < dist[v.first])
            {
                dist[v.first] = new_dist;
                pq.push({new_dist, v.first});
            }
        }
    }
}

void a_star(int start, int goal)
{
    priority_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> pq;
    map<int, int> parent;
    for (int i = 1; i <= n; i++)
    {
        dist[i] = INF;
        heuristic[i] = INF;
    }
    dist[start] = 0;

```

```

pq.push({0, start});
while (!pq.empty())
{
    int u = pq.top().second;
    pq.pop();
    if (u == goal)
        break;
    for (auto v : adj[u])
    {
        int new_dist = dist[u] + v.second;
        if (new_dist < dist[v.first])
        {
            dist[v.first] = new_dist;
            heuristic[v.first] = sqrt(pow(v.first - e, 2) + pow(v.second - e, 2)); // Euclidean distance a
            parent[v.first] = u;
            pq.push({dist[v.first] + heuristic[v.first], v.first});
        }
    }
}

vector<int> path;
int u = goal;
while (u != start)
{
    path.push_back(u);
    u = parent[u];
}
path.push_back(start);
reverse(path.begin(), path.end());
cout << "Shortest path: ";
for (auto u : path)
    cout << u << " ";
cout << endl;
}

int main()
{
    fast;
    cin >> n >> m;
    int u, v, w;
    for (int i = 0; i < m; i++)
    {
        cin >> u >> v >> w;
        adj[u].push_back({v, w});
        adj[v].push_back({u, w});
    }

    cin >> s >> e;

    dijkstra(s);
    a_star(s, e);
}

```