

Name : Vishal Kanhaiya Jha

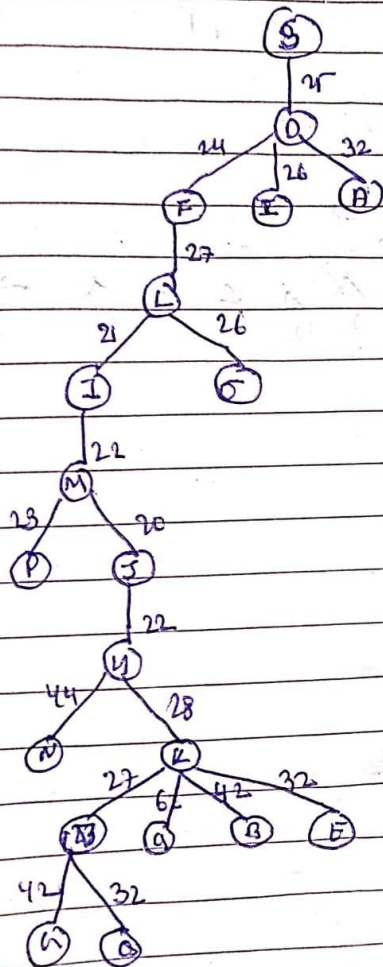
AI / ML Lab

Assignment No 2

T7 Batch

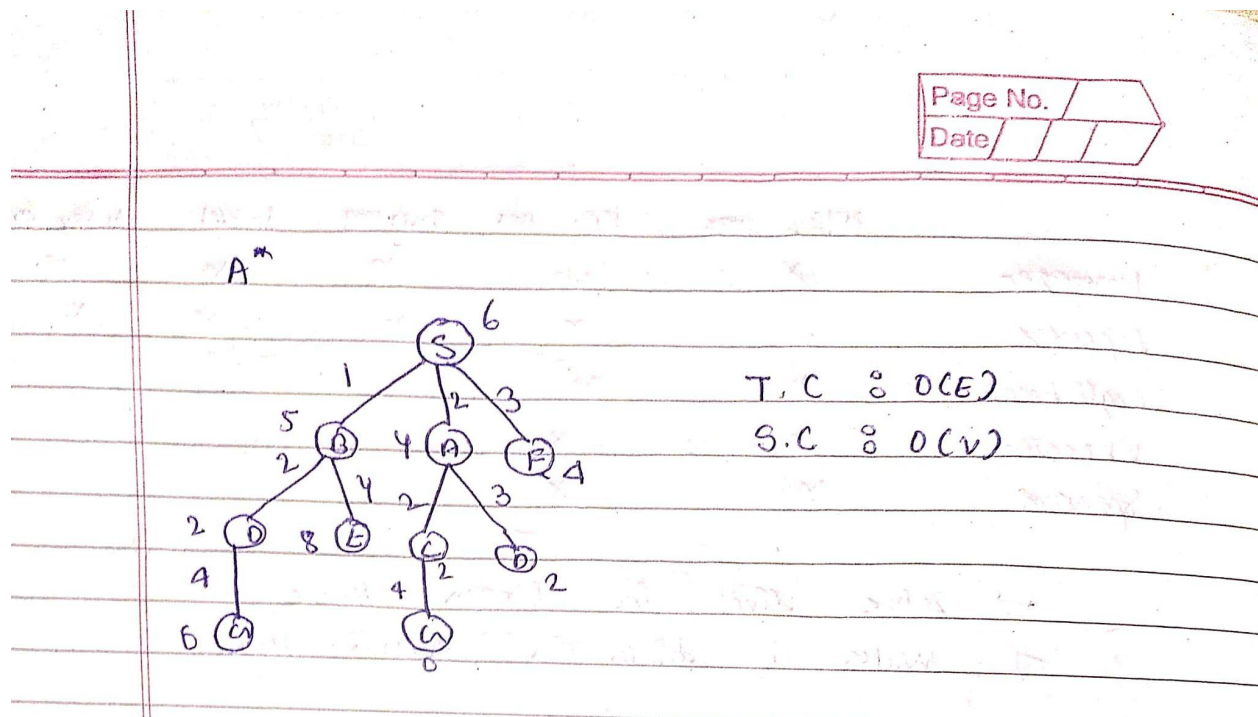
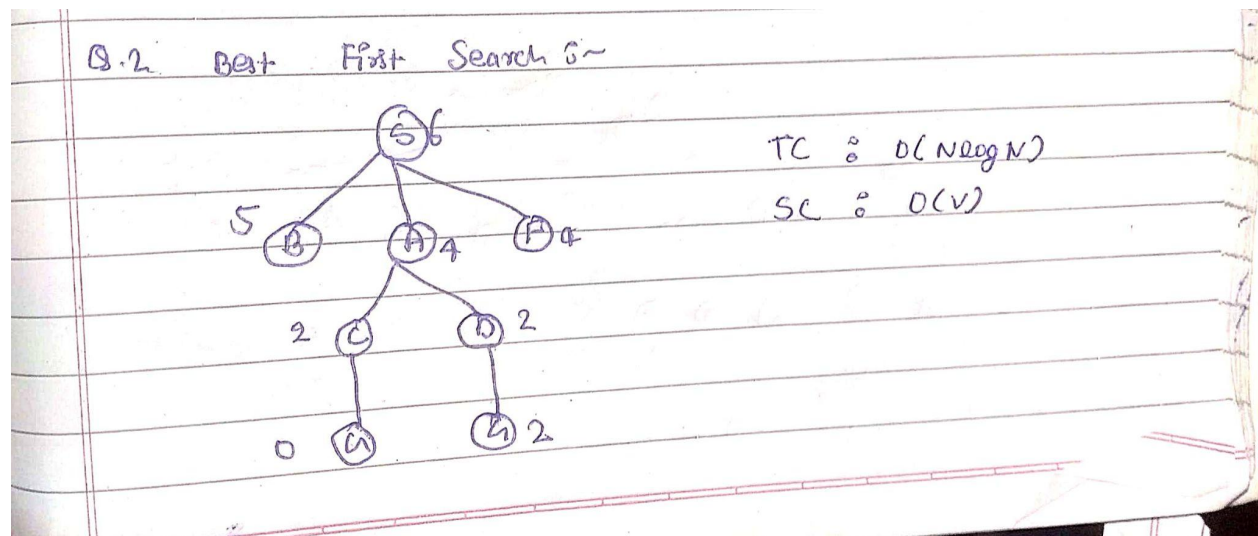
Q. 1)

Q.1 $S(0,0), F(2,0), I(3,1), L(6,3), A(0,3), O(5,1), P(7,4),$
 $I(3,3), N(4,6), K(2,6)$



Path : $S \rightarrow O \rightarrow F \rightarrow L \rightarrow I \rightarrow$
 $M \rightarrow T \rightarrow H \rightarrow K \rightarrow N \rightarrow G$

Q.2)



Q. 3) Not able to solve

Q.4)

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
```

```

#define mod 1000000007

#define N 4

void file()
{
#ifdef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
}

ll bnpow(ll a, ll b)
{
    ll ans = 1;
    while(b > 0)
    {
        if((b & 1) == 1) ans *= a;
        a *= a;
        b = b >> 1;
    }
    return ans;
}

ll gcd(ll a, ll b)
{
    if(b == 0) return a;
    return gcd(b, a%b);
}

ll lcm(ll a, ll b)
{
    return (a / gcd(a, b)) * b;
}

struct Node {
    vector<pair<ll, ll>> path;
    ll rm[N][N];
    ll c;
    ll v;
}

```

```

    ll l;
};

Node* newNode(ll pm[N][N], vector<pair<ll, ll>> const& path, ll l, ll i,
ll j)
{
    Node* node = new Node;
    node->path = path;

    if(l != 0) node->path.push_back({i, j});

    memcpy(node->rm, pm,
sizeof node->rm);

    for (ll k = 0; l != 0 && k < N; k++) {
        node->rm[i][k] = INT_MAX;
        node->rm[k][j] = INT_MAX;
    }

    node->rm[j][0] = INT_MAX;
    node->l = l;
    node->v = j;
    return node;
}

ll rowReduction(ll rm[N][N], ll row[N])
{
    fill_n(row, N, INT_MAX);

    for(ll i = 0; i < N; i++) {
        for(ll j = 0; j < N; j++) {
            if(rm[i][j] < row[i]) {
                row[i] = rm[i][j];
            }
        }
    }

    for (ll i = 0; i < N; i++) {
        for (ll j = 0; j < N; j++) {
            if (rm[i][j] != INT_MAX && row[i] != INT_MAX) {

```

```

        rm[i][j] -= row[i];
    }

}

}

return 0;
}

11 columnReduction(11 rm[N][N], 11 col[N]) {
    fill_n(col, N, INT_MAX);

    for (11 i = 0; i < N; i++) {
        for (11 j = 0; j < N; j++) {
            if (rm[i][j] < col[j]) {
                col[j] = rm[i][j];
            }
        }
    }

    for (11 i = 0; i < N; i++) {
        for (11 j = 0; j < N; j++) {
            if (rm[i][j] != INT_MAX && col[j] != INT_MAX) {
                rm[i][j] -= col[j];
            }
        }
    }

    return 0;
}

11 total(11 rm[N][N]) {
    11 c = 0;
    11 row[N];
    rowReduction(rm, row);
    11 col[N];
    columnReduction(rm, col);

    for (11 i = 0; i < N; i++) {
        c += (row[i] != INT_MAX) ? row[i] : 0;
        c += (col[i] != INT_MAX) ? col[i] : 0;
    }

    return c;
}

```

```

}

struct minHeap {
    bool operator()(const Node* lhs, const Node* rhs) const {
        return lhs->c > rhs->c;
    }
};

ll solve(ll graph[N][N]) {
    priority_queue<Node*, vector<Node*>, minHeap> pq;
    vector<pair<ll, ll> > v;

    Node* root = newNode(graph, v, 0, -1, 0);

    root->c = total(root->rm);

    pq.push(root);

    while (!pq.empty()) {

        Node* min = pq.top();

        pq.pop();

        ll i = min->v;

        if(min->l == N - 1) {
            min->path.push_back(make_pair(i, 0));
            return min->c;
        }

        for(ll j = 0; j < N; j++) {
            if(min->rm[i][j] != INT_MAX) {
                Node* child = newNode(min->rm, min->path, min->l + 1, i,
j);

                child->c = min->c + min->rm[i][j] + total(child->rm);
                pq.push(child);
            }
        }
    }
}

```

```

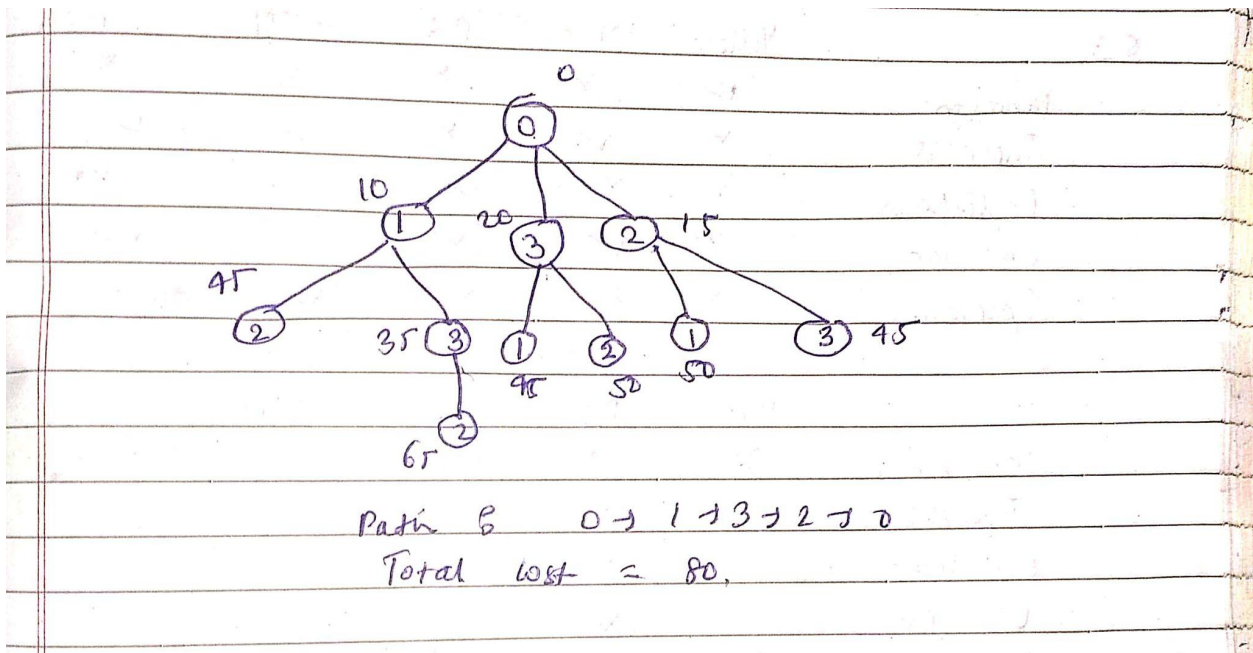
        delete min;
    }
    return 0;
}

void solve()
{
    ll graph[N][N] = { { INT_MAX, 10, 15, 20 },
                        { 10, INT_MAX, 35, 25 },
                        { 15, 35, INT_MAX, 30 },
                        { 20, 25, 30, INT_MAX } };

    cout << endl << "Total Cost : " << solve(graph)<<endl;
}

int main()
{
    file();
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int t = 1;
    // cin >> t;
    while(t--)
    {
        solve();
    }
    return 0;
}

```

Q. 5)

