

Heaven's Light is Our Guide



Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Course No: CSE.2202

Course Title: Sessional based on CSE.2201 (Computer Algorithm)

Report On: Lab Final Problem 1

Submitted To

Dr. Md. Ali Hossain, Associate Professor, Dept. of CSE, RUET

Biprodip Pal, Assistant Professor, Dept. of CSE, RUET

Submitted By

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

Date: 09-08-2021

Problem 1:

Algorithm:

```

① QuickSort (left, right) {
    if (left < right) {
        j = Partition (left, right+1);
        QuickSort (left, j-1);
        QuickSort (j+1, right);
    }
}

```

```

② Partition (m, p) {
    v := a[m]; i = m; j = p;
    repeat {
        repeat { i := i + 1; } until (a[i] >= v);
        repeat { j := j - 1; } until (a[j] <= v);
        if (i < j) then t := a[i]; a[i] = a[j]; a[j] = t;
    } until (i >= j);
    a[m] := a[j]; a[j] = v;
    return j;
}

```

Code :

```
#include <bits/stdc++.h>
using namespace std;
using namespace std::chrono;
typedef long long ll;
#define Max 1e17

vector<ll> arr;
ll n;

void input (ll m; ll i) {
    string a,b = "quick";
    b = b + to_string(i) + ".txt";
    ifstream f1;
    f1.open(b);
    arr. clear();
    while (arr.size() < n) {
        f1 >> a;
        arr.push_back(stod(a));
    }
    arr.push_back(Max);
    f1. close();
}

}
```

```

ll Partition ( ll m, ll p ) {
    ll v = arr[m] ; i = m ; j = p ;
    while ( i < j ) {
        do { i += 1; } while ( arr[i] <= v );
        do { j -= 1; } while ( arr[j] > v );
        if ( i < j ) { swap ( arr[i] , arr[j] ); }
    }
    arr[m] = arr[j] ; arr[j] = v ;
    return j ;
}

void quick_sort ( ll left, ll right ) {
    if ( left < right ) {
        ll j ;
        j = Partition ( left, right + 1 );
        quick_sort ( left, j - 1 );
        quick_sort ( j + 1, right );
    }
}

```

```
int main() {
    ll i;
    for (i = 1; i < 4; i++) {
        cout << "Enter N: ";
        cin >> n;
        input(n, i);
        auto start = high_resolution_clock::now();
        quick_sort(0, n - 1);
        auto stop = high_resolution_clock::now();
        auto duration = duration_cast<milliseconds>(stop - start);
        cout << "Time for " << n << ": " << duration << endl;
    }
    return 0;
}
```

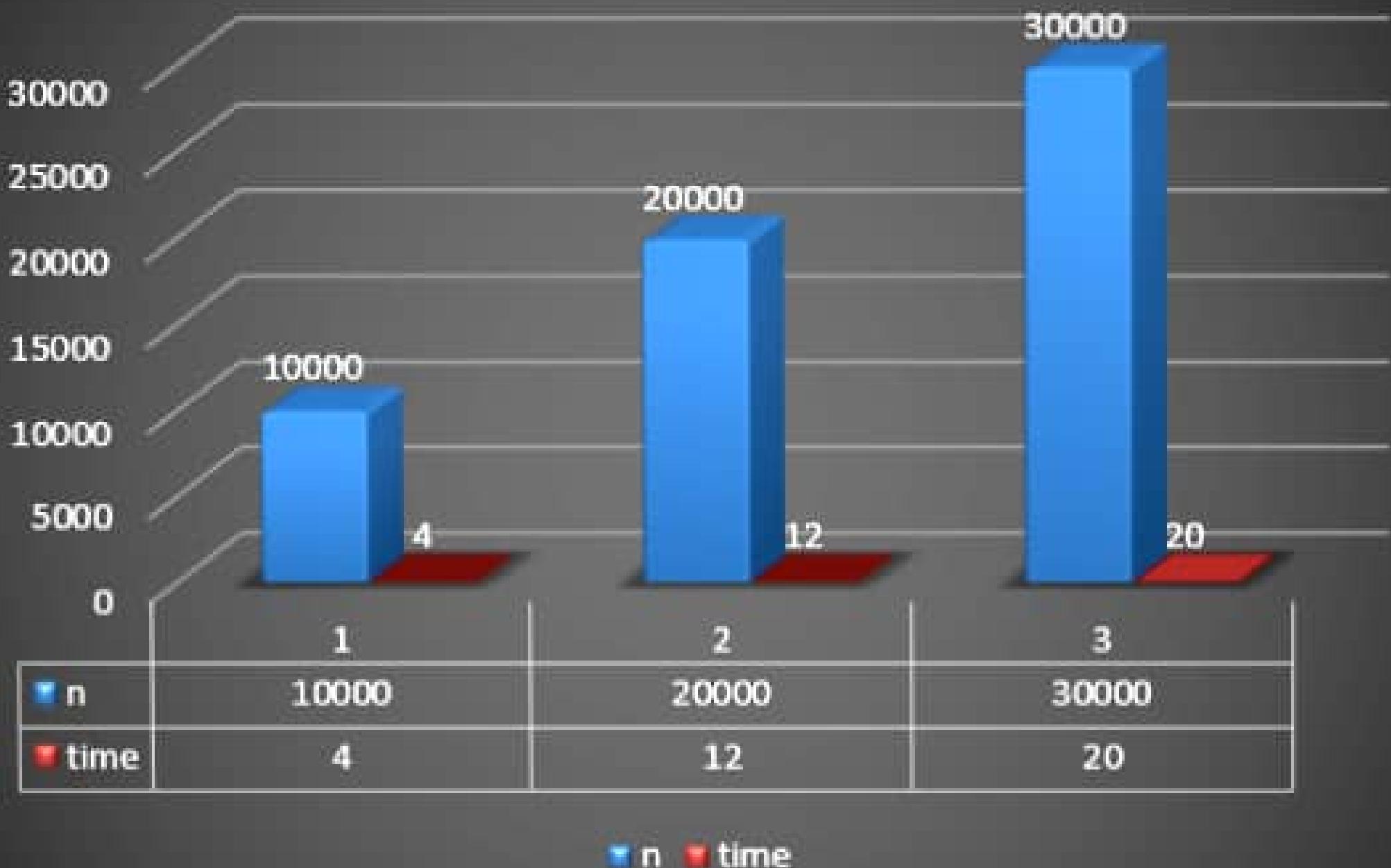
F:\4th Semester\CSE\CSE2202\Lab Final\1\1.exe

Enter Number of Element_(from 5000 to 50000): 10000
Time for 10000 : 4 milliseconds

Enter Number of Element_(from 5000 to 50000): 20000
Time for 20000 : 12 milliseconds

Enter Number of Element_(from 5000 to 50000): 30000
Time for 30000 : 20 milliseconds

process returned 0 (0x0) execution time : 20.962 s
Press any key to continue.



Heaven's Light is Our Guide



Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Course No: CSE.2202

Course Title: Sessional based on CSE.2201 (Computer Algorithms)

Report On: Lab Final Problem 2

Submitted To

Dr. Md. Ali Hossain, Associate Professor, Dept. of CSE, RUET

Biprodip Pal, Assistant Professor, Dept. of CSE, RUET

Submitted By

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

Date: 09-08-2021

Problem 2:

Algorithm:

```
① MergedSort (low, high){  
    if (low < high){  
        mid := ⌊(low+high)/2⌋;  
        MergedSort (low, mid);  
        MergedSort (mid+1, high);  
        Merge (low, mid, high);  
    }  
}
```

```
② Merge (low, mid, high){  
    i := low, j := mid+1; k := high;  
    while (i ≤ mid) and (j ≤ high) do {  
        if (a[i] ≤ a[j]) then {  
            b[i] := a[i]; i := i+1;  
        } else { b[i] = a[j]; j := j+1; }  
        i := i+1;  
    }  
    if (i > mid) then
```

```

for k:= h to high do {
    b[i] = a[k]; i = i+1;
}
else { for k := h to mid do {
    b[i] = a[k]; i = i+1;
}
}
for k = low to high do a[k] = b[k];
}

```

Code :

```

#include <bits/stdc++.h>
using namespace std;
using namespace std::chrono;
typedef long long ll;
vector<ll> arr1;
ll arr2[100000], n;

```

```

void input (ll n, ll i) {
    string a, b = "mange";
    b = b + to_string(i) + ".txt";
    ifstream f1;

```

```

f1.open(b);
arr1.clean();
while (arr1.size() < n) {
    f1 >> a;
    arr1.push_back(stoi(a));
}
f1.close();

}

void merge (ll low, ll mid, ll high) {
    ll h=low, i=low, j=high;
    while (h <= mid && j <= high) {
        if (arr1[h] <= arr1[j]) {
            arr2[i] = arr1[h];
            h++;
            i++;
        } else {
            arr2[i] = arr1[j];
            j++;
        }
    }
    if (h > mid) {
        for (ll k=j; k <= high; k++) {
            arr2[i] = arr1[k];
            i++;
        }
    }
}

```

```
else { for (ll k = h; k <= mid; k++) {  
    arr2[i] = arr1[k];  
    i++; } }
```

```
for (ll i = low; i <= high; i++) {  
    arr1[i] = arr2[i]; } }
```

```
void merge_sort (ll low, ll high) {  
    if (low < high) {  
        ll mid = (low + high) / 2;  
        merge_sort (low, mid);  
        merge_sort (mid + 1, high);  
    } }
```

```
int main() {
    int i;
    for (i = 1; i < 4; i++) {
        cout << "Enter N : "; cin >> n;
        input(n, i);
        auto start = high_resolution_clock::now();
        range_sort(0, n - 1);
        auto stop = high_resolution_clock::now();
        auto duration = duration_cast<millisecond>(stop - start);
        cout << "\nTime for " << n << ": " << duration << endl;
    }
    return 0;
}
```



"F:\4th Semester\CSE\CSE.2202\Lab Final\2\2.exe"

Enter Number of Element_(from 5000 to 50000): 10000

Time of 10000 : 11 milliseconds

Enter Number of Element_(from 5000 to 50000): 20000

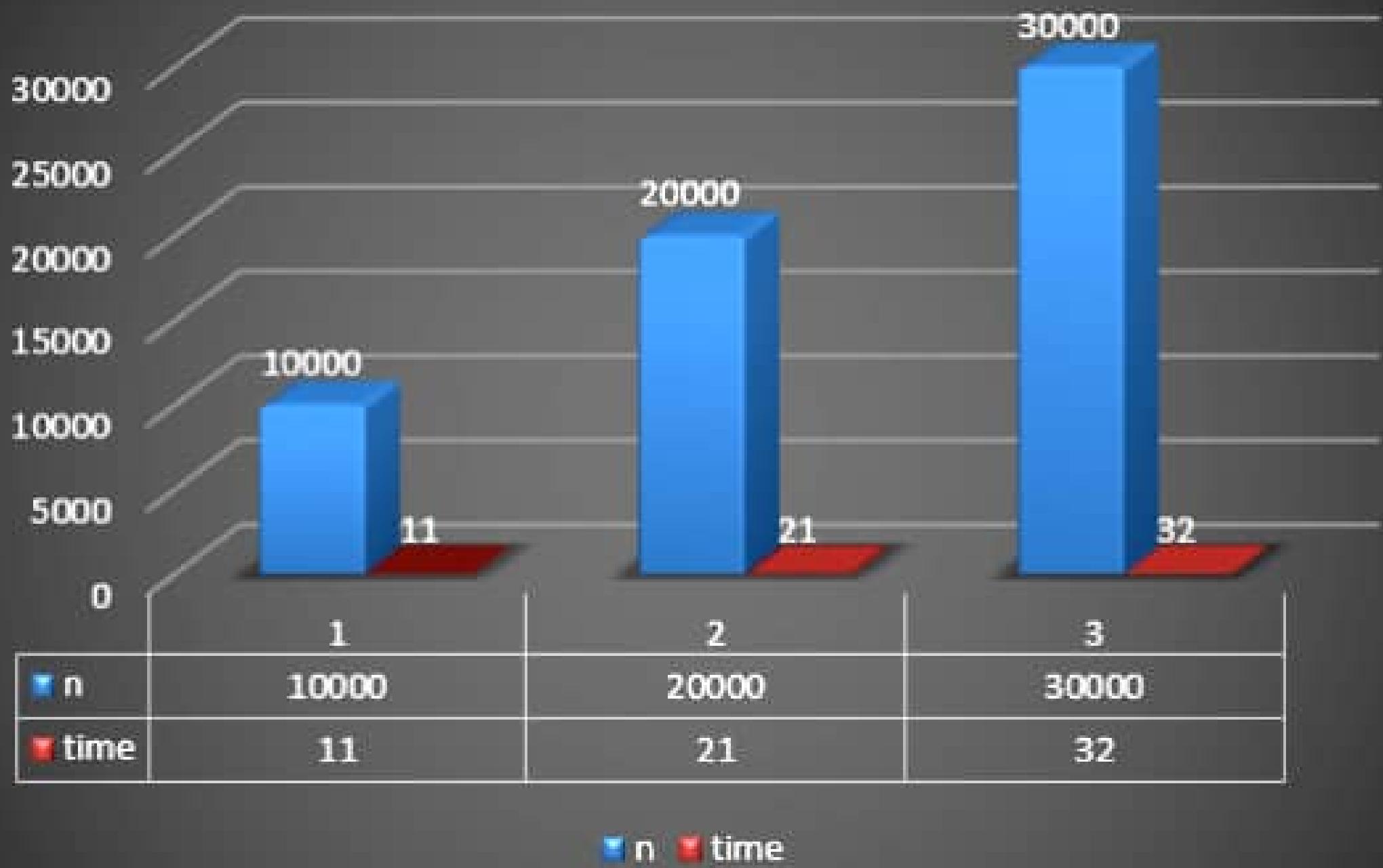
Time of 20000 : 21 milliseconds

Enter Number of Element_(from 5000 to 50000): 30000

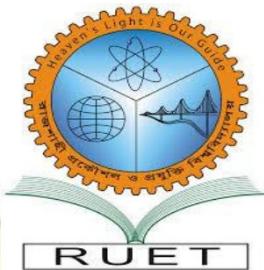
Time of 30000 : 32 milliseconds

Process returned 0 (0x0) execution time : 11.738 s

Press any key to continue.



Heaven's Light is Our Guide



Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Course No: CSE.2202

Course Title: Sessional based on CSE.2201 (Computer Algorithms)

Report On: Lab Final Problem 3

Submitted To

Dr. Md. Ali Hossain, Associate Professor, Dept. of CSE, RUET

Biprodip Pal, Assistant Professor, Dept. of CSE, RUET

Submitted By

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

Date: 09-08-2021

Problem 3:

Algorithm:

① Greedy knapsack (m, n) {

// object are orden as $p[i]/w[i] \geq p[i+1]/w[i+1]$

// m is the size of knap sack.

// n is the number of objects.

for $i := 1$ to n do $w[i] = 0.0$

$U := m$

for $i := 1$ to n do {

if ($w[i] > U$) then break;

$w[i] = 1.0$; $U = U - w[i]$;

}

if ($i \leq n$) then $w[i] := U/w[i]$;

}

⑥ 0/1 knapsack (w , n) {

for $i = 0$ to w do {

for $j = 0$ to n do {

if ($i \geq 0$ and $j \geq 0$) then

$dp[i][j] = 0;$

else if ($i - w[j-1] < 0$) then

$dp[i][j] = dp[i][j-1];$

else {

$dp[i][j] = \max(dp[i][j-1],$
 $(dp[i - w[j-1]][j] + v[j-1]));$

}

}

}

}

Code :

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

int cur = 11; cur1, n=0;
float p=0, dp[200][200];
vector<float> nwp[5];

void inputs ( int cur[], int v[] ) {
    for ( int i=0; i<5; i++ ) {
        nwp[0].push_back ( i+1 );
        nwp[1].push_back ( cur[i] );
        nwp[2].push_back ( v[i] );
    }
}

for ( int i=0; i<5; i++ ) {
    nwp[3].push_back ( nwp[2][i]/nwp[1][i] );
}

for ( int i=0; i<5; i++ ) {
    for ( int j=i+1; j<5; j++ ) {
        if ( nwp[3][i] < nwp[3][j] ) {
            swap ( nwp[0][i], nwp[0][j] );
            swap ( nwp[1][i], nwp[1][j] );
            swap ( nwp[2][i], nwp[2][j] );
        }
    }
}
```

```

        swap(nmp[2][i], nmp[2][j]);
        swap(nmp[3][i], nmp[3][j]);
    }
}

}

}

void fractionalknapsack() {
    cout << "Fractional knapsack : " << p << endl;
}

```

```

void 0_1_knapsack( int w[], int v[] ) {
    for ( int i=0; i<=cut; i++ ) {
        for ( int j=0; j<=5; j++ ) {
            if ( i==0 || j==0 ) {
                dp[i][j] = 0;
            }
            else if ( i-w[j-1] < 0 ) {
                dp[i][j] = dp[i][j-1];
            }
            else {
                dp[i][j] = max( dp[i][j-1], ( dp[i-w[j-1]][j-1]
                    + v[j-1] ) );
            }
        }
    }
    cout << " 0/1 knapsack: " << dp[cut][5] << endl;
}

```

```
int main() {  
    int w[] = { 1, 2, 5, 6, 7 };  
    int v[] = { 1, 6, 18, 22, 28 };  
  
    inputs(w, v);  
  
    fractional_knapsack();  
    2_0_knapsack();  
  
    return 0;  
}
```

F:\4th Semester\CSE\CSE.2202\Lab Final\3\3.exe

Item	Weight	Profit	P/W
1	1	1	1
2	2	6	3
3	5	18	3.6
4	6	22	3.66667
5	7	28	4

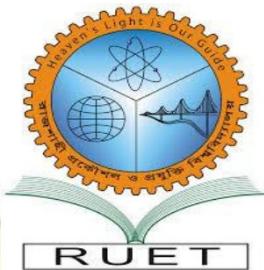
Fractional Knapsack: 42.6667

0/1 Knapsack: 40

Process returned 0 (0x0) execution time : 0.253 s

Press any key to continue.

Heaven's Light is Our Guide



Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Course No: CSE.2202

Course Title: Sessional based on CSE.2201 (Computer Algorithms)

Report On: Lab Final Problem 4

Submitted To

Dr. Md. Ali Hossain, Associate Professor, Dept. of CSE, RUET

Biprodip Pal, Assistant Professor, Dept. of CSE, RUET

Submitted By

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

Date: 09-08-2021

Problem 4.

Algorithm :

① Counting sort (data n[]){

for i = 0 to n do {

 element := data[i];

 F[element] += 1;

}

for i = 1 to n do {

 P[i] = F[i] + F[i-1];

}

for i = 1 to n {

 Element = data[i];

 S-A[P[Element]] = Element;

}

}

```

⑩ BubbleSort (data n[]){
    for i=0 to n do {
        for j=0 to n do {
            if (data[i] > data[j]){
                swap (data[i],data[j]);
            }
        }
    }
}

```

Code:

```

#include <iostream.h>
using namespace std;
using namespace std::chrono;
typedef long long ll;
#define M 10001

```

```
void menu() {
    cout << "Enter N (0 to exit): ";
}

int main() {
    ll n, i, j;
    vector<ll> cn, cb, cc;
    while (1) {
        ll a;
        menu();
        cin >> a;
        if (a < 0) { cout << "Invalid input" << endl; continue }
        if (a == 0) { break; }

        else
        ll mn, mn, n;
        vector<ll> bsort_array, pos_ssort, crnt_array;
        map<ll, ll> mp;
        ifstream f1;
        ifstream f2;
```

```
f1.open ("input.txt");
n = a;
en.push_back(n);
snand(time(0));
n = rand() % M;
mn = n; mn = n;
f1 << n;
esort_array.push_back(-1);

for(i = 1; i < n; i++) {
    n = rand()%M;
    f1 << " "; f1 << n;
    mx = max(mn, n);
    mn = min(mn, n);
    esort_array.push_back(-1);
}

f1.close();

auto s = high_resolution_clock::now();
f2.open ("output.txt");
while (!f2.eof()) {
    f2 >> x;
    esort_array.push_back(x);
}
```

```

f2.close();

for (i=0; i<n-1; i++) {
    for (j=i+1; j<n; j++) {
        if (bsort_array[i] > bsort_array[j]) {
            swap(bsort_array[i], bsort_array[j]);
        }
    }
}

auto st = high_resolution_clock::now();
auto dur = duration_cast<milliseconds>(st - s);
cout << "Time Bubble Sort :" << dur << endl;
cb.push_back(dur.count());

```

```

s = high_resolution_clock::now();
f2.open("input-bits");
while (!f2.eof()) {
    f2 >> n;
    mp[n] += 1;
}

```

```
f2.close();
```

```
for (i = min; i <= max; i++) {
```

```
    if (i == mn) {
```

```
        pos_csort.push_back(mp[i]);
```

```
        continue;
```

```
}
```

```
        pos_csort.push_back(mp[i] + pos_csort[i - mn - 1]);
```

```
}
```

```
f2.open("input.txt");
```

```
while (!f2.eof()) {
```

```
f2 >> n;
```

```
csort_array[pos_csort[n - mn] - 1] = n;
```

```
pos_csort[n - mn] -= 1;
```

```
}
```

```
f2.close();
```

```
st = high_resolution_clock::now();
```

```
dur = duration_cast<milliseconds>(st - s);
```

```
cout << "Counting sort: " << dur.count() << endl;
```

```
} return 0;
```

Enter N (Press 0 to Exit): 100

Bubble Sort: 8 Milliseconds

Counting Sort: 16 Milliseconds

Enter N (Press 0 to Exit): 400

Bubble Sort: 13 Milliseconds

Counting Sort: 19 Milliseconds

Enter N (Press 0 to Exit): 700

Bubble Sort: 22 Milliseconds

Counting Sort: 21 Milliseconds

Enter N (Press 0 to Exit): 1000

Bubble Sort: 27 Milliseconds

Counting Sort: 16 Milliseconds

Enter N (Press 0 to Exit): 1300

Bubble Sort: 33 Milliseconds

Counting Sort: 14 Milliseconds

Enter N (Press 0 to Exit): 1600

Bubble Sort: 42 Milliseconds

Counting Sort: 15 Milliseconds

Enter N (Press 0 to Exit): 1900

Bubble Sort: 40 Milliseconds

Counting Sort: 15 Milliseconds

Enter N (Press 0 to Exit): 2200

Bubble Sort: 59 Milliseconds

Counting Sort: 15 Milliseconds

Enter N (Press 0 to Exit): 2500

Bubble Sort: 68 Milliseconds

Counting Sort: 16 Milliseconds

Enter N (Press 0 to Exit): 0

Exiting...

N	B_S	C_S
100	8	16
400	13	19
700	22	21
1000	27	16
1300	33	14
1600	42	15
1900	40	15
2200	59	15
2500	68	16

Heaven's Light is Our Guide



Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Course No: CSE.2202

Course Title: Sessional based on CSE.2201 (Computer Algorithms)

Report On: Lab Final Problem 5

Submitted To

Dr. Md. Ali Hossain, Associate Professor, Dept. of CSE, RUET

Biprodip Pal, Assistant Professor, Dept. of CSE, RUET

Submitted By

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

Date: 09-08-2021

Problem 5 :

Algorithm :

① BFS (node) {

 for i = 0 to n do {

 if (node[x][i] = 1) { then

 enqueue (i);

}

}

 while (queue is not empty) {

 x = queue front();

 dequeue();

 if x is not visited then {

 print x;

 visit x;

}

 BFS (x);

}

}

Code :

```
#include <iostream>
using namespace std;
typedef long long ll;
queue<int> q;
int gp[105][105] = {0};
int vis[105] = {0};

void bfs (int n, int v) {
    for (int i = 1; i <= v; i++) {
        if (gp[n][i] == -1 && vis[i] != 1) {
            q.push(i);
        }
    }
    while (!q.empty()) {
        int u = q.front();
        q.pop();
        if (vis[u] != 1) {
            cout << "x" << endl;
            vis[u] = 1;
        }
        bfs(u);
    }
}
```

```

int main() {
    int v, i, j, k;
    cout << "Enter the Number of vertices: ";
    cin >> v;
    cout << "Enter graph data in matrix form: " << endl;
    for (i=1; i<=v; i++) {
        for (j=1; j<=v; j++) {
            cin >> gp[i][j];
        }
    }
    cout << "Enter the start node: "; cin >> k;
    vis[k] = 1;
    cout << "\nThe nodes which are reachable are: ";
    bfs(k, v);
    cout << endl;
    return 0;
}

```

"F:\4th Semester\CSE\CSE.2202\Lab Final\5\5.exe"

Enter the Number of vertexes: 4

Enter graph data in matrix form:

0 1 1 1
0 1 0 0
0 0 0 0
0 0 1 0

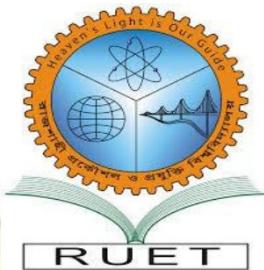
Enter the starting vertex: 1

The node which are reachable are: 2 3 4

Process returned 0 (0x0) execution time : 42.691 s

Press any key to continue.

Heaven's Light is Our Guide



Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Course No: CSE.2202

Course Title: Sessional based on CSE.2201 (Computer Algorithms)

Report On: Lab Final Problem 6

Submitted To

Dr. Md. Ali Hossain, Associate Professor, Dept. of CSE, RUET

Biprodip Pal, Assistant Professor, Dept. of CSE, RUET

Submitted By

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

Date: 09-08-2021

Problem 6 :

Algorithm :

i) IsSafe (k, i) {

for (j := 1 to k-1) {

if ($x[j] = i$) or

$\text{abs}(x[j] - i) = \text{abs}(j - k)$ then,

return false ;

}

return true;

}

ii) NQueens (k, n) {

for (i := 1 to n) {

if (IsSafe) then {

$x[k] = i$;

if ($k = n$) print output;

else NQueen (k+1, n); }

}

7

Code:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
typedef long long ll;
```

```
int arr[50] = {0};
```

```
vector<int> v;
```

```
bool issafe(int n, int y, int &n) {
```

```
    bool result = true;
```

```
    int f = 1;
```

```
    for (int k = n - 1; k > 0; k--) {
```

```
        if (arr[k] == y) { result = false;
```

```
            break;
```

```
        } if ((n + y) == arr[k] + k) && y < n) {
```

```
            result = false;
```

```
            break;
```

```
}
```

```
if (y > 1 && arr[y] == (y - f)) {
```

```
    result = false;
```

```
    break;
```

```
} f += 1;
```

```
}
```

return result;

}

```
void NQ( int a, int &n ) {  
    for ( int i = 1; i <= n; i++ ) {  
        if ( isafe(a, i, n) ) {  
            arr[a] = i;  
            if ( a == n ) {  
                cout << "Solutions:" << endl;  
                for ( int j = 1; j <= n; j++ ) {  
                    for ( int k = 1; k <= n; k++ ) {  
                        if ( arr[j] == k ) {  
                            cout << "Q ";  
                        }  
                        else { cout << "* "; }  
                    }  
                    cout << endl;  
                }  
                else { NQ(a+1, n) }  
            }  
        }  
    }  
}
```

```
int main() {
    int n;
    cout << "Enter N: ";
    cin >> n;
    if (n >= 4) {
        NQ(1,
        NQ(1, n);
    }
}
```

return 0;

}

F:\4th Semester\CSE\CSE2202\Lab Final\6\6.exe

Enter N: 4

Solution:

```
*   Q   *   *
*   *   *   Q
Q   *   *   *
*   *   Q   *
```

Solution:

```
*   *   Q   *
Q   *   *   *
*   *   *   Q
*   Q   *   *
```

Iteration: 75

Process returned 0 (0x0) execution time : 7.333 s
Press any key to continue.