**RBC measure on Dijkstra Algorithm implementation with C# Language**

Given a Graph G (V, E)



**Fig. 3.1.**

**Input**

Extracting from the graph in fig. 3.1 above:

N = Number of vertices = 6 ⇒ Size (n) = 1

V = Vertices defined as {V1, V2, …, Vn} = { 1, 2, 3, 4, 5, 6 } ⇒ Size (V) = 6

M = Number of Edges = 9 ⇒ Size (M) = 1

E = Edges defined as {Weight (u, v): u, v E V} = { 7, 9,10, 14, 2, 11, 15, 9, 6}

= Size (E) = 9

A = Source vertex = 1, ⇒ size (a) = 1

B = Target vertex = 5, ⇒ size (b) = 1

Dist = Distance = {0, ∞, ∞, ∞, ∞, ∞}, ⇒ Size (Dist) = 6

Number of Input = Size (n) + Size(V) + Size (m) + Size (E) + Size ( a ) + Size (b) + Size (Dist) = 1 + 6 + 1 + 9 + 1 + 1 + 6 = 25

**Output**

The shortest path form the source a to the target b is given as

Prev = { 1, 3, 6, 5}, size (prev) = 4

The list of the shortest path, which is given as

C = 9 + 2 + 9 = 20, size (c ) = 1

Number of output = Size (prev) + Size (c ) = 4 + 1 = 5

Number of Interface = 1 (console windows)

Number of files = 1 ( for data storage)

**1. Input Output Complexity (IOC)**

IOC = No of Input + No of Output + No of interfaces + No of files

= 25 + 5 + 1 + 1 = 32

**2. Functional Requirement (FR)**

No of function ⇒ (DijkstraShortestPath ( ), SetNoOfNodes ( ), GetNoOfNodes ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), FindShortestPath ( ), minDist ( ) ) = 9

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- process such as Arithmetic, Computation, Display, etc.

Function 1 - ShortestPath ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 2 - ReadNodes ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 3 - ReadWeight ( ) has 3 (Display, Read, Type-casting operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - FindShortestPath ( ) has 2 (Display, arithmetic addition)

= 3 + 3 + 3 + 1 + 2 = 12

FR = Number of functions \* = 9 \* 12 = 108

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 108 + 0 = 108

**5. Product Complexity (PC)**

PC = IOC \* RC = 32 \* 108 = 3456

**6**. **Personal Complexity Attributes (PCA)**

PCA = 1.17 (Suppose Programmer Capability = Low)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((3456 \* 1.17) + 0 + 0 + 0 ) \* 1 = 4043.52

**RBC measure on Dijkstra Algorithm implementation with C++ Language**

**1. Input Output Complexity (IOC)**

Number of Input = Size (n) + Size(V) + Size (m) + Size (E) + Size ( a ) +

Size (b) + Size (Dist) = 1 + 6 + 1 + 9 + 1 + 1 + 6 = 25

Number of output = Size (prev) + Size (c) = 4 + 1 = 5

Number of Interface = 1 (console window)

Number of files = 1 (for data storage)

IOC = No of Input + No of Output + No of interfaces + No of files

= 25 + 5 + 1 + 1 = 32

**2. Functional Requirement (FR)**

No of functions ⇒ (DijkstraShortestPath ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), minDist ( )) = 6

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- process such as Arithmetic, Computation, Display, etc.

Function 1 - ShortestPath ( ) has 2 (Display operations, Read operations )

Function 2 - ReadNodes ( ) has 2 (Display operations, Read operations)

Function 3 - ReadWeight ( ) has 2 (Display operations, Read operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - DijkstraShortestPath ( ) has 2 (Display, arithmetic addition)

= 2 + 2 + 2 + 1 + 2 = 9

FR = Number of functions \* = 6 \* 9 = 54

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 54 + 0 = 54

**5. Product Complexity (PC)**

PC = IOC \* RC = 32 \* 54 = 1728

**6**. **Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((1728 \* 0.90) + 0 + 0 + 0 ) \* 1 = 1555.2

**RBC measure on Dijkstra Algorithm implementation with VB 6.0 Language**

**1. Input Output Complexity (IOC)**

Number of Input = Size (n) + Size(V) + Size (m) + Size (E) + Size ( a ) +

Size (b) + Size (Dist) = 1 + 6 + 1 + 9 + 1 + 1 + 6 = 25

Number of output = Size (prev) + Size (c) = 4 + 1 = 5

Number of Interface = 1 (console window)

Number of files = 1 (for data storage)

IOC = No of Input + No of Output + No of interfaces + No of files

= 25 + 5 + 1 + 1 = 32

**2. Functional Requirement (FR)**

No of functions ⇒ (ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), FindShortestPath ( ), minDist ( )) = 5

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- process such as Arithmetic, Computation, Display, etc.

Function 1 - ReadNodes ( ) has 2 (Display operations, Read operations)

Function 2 - ReadWeight ( ) has 2 (Display operations, Read operations)

Function 3 - DisplayWeight ( ) has 1 (Display operations)

Function 4 - FindShortestPath ( ) has 2 (Display operations, arithmetic addition)

= 2 + 2 + 1 + 2 = 7

FR = Number of functions \* = 5 \* 7 = 35

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 35 + 0 = 35

**5. Product Complexity (PC)**

PC = IOC \* RC = 32 \* 35 = 1120

**6**. **Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((1120 \* 0.90) + 0 + 0 + 0 ) \* 1 = 1008

**RBC measure on Dijkstra Algorithm implementation with Java Language**

**1. Input Output Complexity (IOC)**

Number of Input = Size (n) + Size(V) + Size (m) + Size (E) + Size ( a ) +

Size (b) + Size (Dist) = 1 + 6 + 1 + 9 + 1 + 1 + 6 = 25

Number of output = Size (prev) + Size (c ) = 4 + 1 = 5

Number of Interface = 1 (console windows)

Number of files = 1 ( for data storage)

IOC = No of Input + No of Output + No of interfaces + No of files

= 25 + 5 + 1 + 1 = 32

**2. Functional Requirement (FR)**

No of functions ⇒ (DijkstraShortestPath ( ), SetNoOfNodes ( ), GetNoOfNodes ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), FindShortestPath ( ), minDist ( )) = 9

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- process such as Arithmetic, Computation, Display, etc.

Function 1 - ShortestPath ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 2 - ReadNodes ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 3 - ReadWeight ( ) has 3 (Display, Read, Type-casting operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - FindShortestPath ( ) has 2 (Display, arithmetic addition)

= 3 + 3 + 3 + 1 + 2 = 12

FR = Number of functions \* = 9 \* 12 = 108

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 108 + 0 = 108

**5. Product Complexity (PC)**

PC = IOC \* RC = 32 \* 108 = 3456

**6**. **Personal Complexity Attributes (PCA)**

PCA = 1.00 (Suppose Programmer Capability = Normal)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1, Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((3456 \* 1.00) + 0 + 0 + 0 ) \* 1 = 3456

**RBC measure on Floyd – Warshall Algorithm implemented with C#**

Given a graph G (V, E)



**Fig. 3.2.**

**Input**

Extracting from the graph in fig. 3.2. above:

N = Number of vertices V = 4, size (N) = 1

V = { V1, V2, V3, V4} = {1, 2, 3, 4} = size (V) = 4

M = Number of Edges = 5, size (M) = 1

E = {w (u, v): u, v € V} = { 4, -2, 2, -1, 3}, size (E) = 5

Dist = |V| x |V| array of minimum distance initialized to (infinity)

Where N = |V| = 4

Size (Dist) = 4 x 4 = 16

Next = |V| \* |V| array of vertex indices initialized to null

Size (next) = 4 x 4 = 16

A = Source vertex = 2, size (a) = 1

B = Target vertex = 3, size (b) = 1

No of input = Size (N) + Size (V) + Size (M) + Size (E) + Size (Dist) + Size (Next)

+ Size (a) + Size (b) = 1 + 4 + 1 + 5 +16 + 16 + 1 + 1 = 45

**Output**

* Dist = |V| x |V| array of the computed minimum or shortest distance from any

Source vertex you to destination vertex V

* The shortest path constructed from a given source a to target vertex b, which is:

Path = { 2, 1, 3}, Size (path) = 3

* The cost/weight of the shortest path computed as; c = 4 + (-2) = 2, size (c ) = 1

No of output = Size (Dist) + Size (Path) + Size (c ) = 16 + 3 + 1 = 20

No of interface = 1 (console window)

No of fites = 1 (Data storage)

1. **Input Output Complexity (IOC)**

IOC = No of input + No of output + No of Interface + No of files

= 45 + 20 + 1 + 1 = 67

1. **Functional Requirement (FR)**

No of functions = (FloydWarshellShortestPath ( ), SetNoOfNodes ( ), GetNoOfNodes ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), ComputeShortestPath ( ), DisplayShortestPath ( ) FindShortestPath ( ) = 10

**No of sub-process/sub- functions**

Function 1 - ShortestPath ( ) has 3 (Display, Read, Type-Casting operations)

Function 2 - ReadNodes ( ) has 3 (Display, Read, Type-Casting operations)

Function 3 - ReadWeight ( ) has 3 (Display, Read, Type-casting operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - ComputeShortestPath( ) has 1 (arithmetic addition)

Function 6 - DisplayShortestPath ( ) 1 (Display operation)

Function 7 - FindShortestPath ( ) 4 (Display, Read, Type-Casting & increment operations)

= 3 + 3 + 3 + 1 + 1 + 1 + 4 = 16

FR = Number of functions \* = 10 \* 16 = 160

**3. Non functional Requirement (NFR)**

NFR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 160 + 0 = 160

**5. Product Complexity (PC)**

PC = IOC \* RC = 67 \* 160 = 10720

**6. Personal Complexity Attributes (PCA)**

PCA = 1.17 (Suppose Programmer Capability = Low)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((10720 \* 1.17) + 0 + 0 + 0 ) \* 1

= 12542.4

**RBC measure on Floyd – Warshall Algorithm implemented with C++**

1. **Input Output Complexity (IOC)**

No of input = Size (N) + Size (V) + Size (M) + Size (E) + Size (Dist) + Size (Next)

+ Size (a) + Size (b) = 1 + 4 + 1 + 5 +16 + 16 + 1 + 1 = 45

No of output = Size (Dist) + Size (Path) + Size (c ) = 16 + 3 + 1 = 20

No of interface = 1 (console window)

No of fites = 1 (Data storage)

IOC = No of input + No of output + No of Interface + No of files

= 45 + 20 + 1 + 1 = 67

1. **Functional Requirement (FR)**

No of functions = (ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), ComputeShortestPath ( ), DisplayShortestPath ( ) FindShortestPath ( )) = 7

**No of sub-process/sub- functions**

Function 1 - ShortestPath ( ) has 2 (Display, and Read Operations)

Function 2 - ReadNodes ( ) has 2 (Display, and Read operations)

Function 3 - ReadWeight ( ) has 2 (Display, and Read operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - ComputeShortestPath( ) has 1 (arithmetic addition)

Function 6 - DisplayShortestPath ( ) has 1 (Display operation)

Function 7 - FindShortestPath ( ) has 3 (Display, Read, and increment operations)

= 2 + 2 + 2 + 1 + 1 + 1 + 3 = 12

FR = Number of functions \* = 7 \* 12 = 84

**3. Non functional Requirement (NFR)**

NFR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 84 + 0 = 84

**5. Product Complexity (PC)**

PC = IOC \* RC = 67 \* 84 = 5628

**6. Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((5628 \* 0.90) + 0 + 0 + 0 ) \* 1 = 5065.2

**RBC measure on Floyd – Warshall Algorithm implemented with VB 6.0**

1. **Input Output Complexity (IOC)**

No of input = Size (N) + Size (V) + Size (M) + Size (E) + Size (Dist) + Size (Next)

+ Size (a) + Size (b) = 1 + 4 + 1 + 5 +16 + 16 + 1 + 1 = 45

No of output = Size (Dist) + Size (Path) + Size (c ) = 16 + 3 + 1 = 20

No of interface = 1 (console window)

No of fites = 1 (Data storage)

IOC = No of input + No of output + No of Interface + No of files

= 45 + 20 + 1 + 1 = 67

1. **Functional Requirement (FR)**

No of functions = (ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), ComputeShortestPath ( ), DisplayShortestPath ( ) FindShortestPath ( )) = 6

**No of sub-process/sub- functions**

Function 1 - ReadNodes ( ) has 2 (Display, and Read operations)

Function 2 - ReadWeight ( ) has 2 (Display, and Read operations)

Function 3 - DisplayWeight ( ) has 1 (Display operations)

Function 4 - ComputeShortestPath( ) has 1 (arithmetic addition)

Function 5 - DisplayShortestPath ( ) has 1 (Display operation)

Function 6 - FindShortestPath ( ) has 3 (Display, Read, and increment operations)

= 2 + 2 + 1 + 1 + 1 + 3 = 10

FR = Number of functions \* = 6 \* 10 = 60

**3. Non functional Requirement (NFR)**

NFR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 60 + 0 = 60

**5. Product Complexity (PC)**

PC = IOC \* RC = 67 \* 60 = 4020

**6. Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1, Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((4020 \* 0.90) + 0 + 0 + 0 ) \* 1 = 3618

**RBC measure on Floyd – Warshall Algorithm implemented with JAVA**

1. **Input Output Complexity (IOC)**

No of input = Size (N) + Size (V) + Size (M) + Size (E) + Size (Dist) + Size (Next)

+ Size (a) + Size (b) = 1 + 4 + 1 + 5 +16 + 16 + 1 + 1 = 45

No of output = Size (Dist) + Size (Path) + Size (c ) = 16 + 3 + 1 = 20

No of interface = 1 (console window)

No of fites = 1 (Data storage)

IOC = No of input + No of output + No of Interface + No of files

= 45 + 20 + 1 + 1 = 67

1. **Functional Requirement (FR)**

No of functions = (FloydWarshellShortestPath ( ), SetNoOfNodes ( ), GetNoOfNodes ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), ComputeShortestPath ( ), DisplayShortestPath ( ) FindShortestPath ( ) = 10

**No of sub-process/sub- functions**

Function 1 - ShortestPath ( ) has 3 (Display, Read, Type-Casting operations)

Function 2 - ReadNodes ( ) has 3 (Display, Read, Type-Casting operations)

Function 3 - ReadWeight ( ) has 3 (Display, Read, Type-casting operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - ComputeShortestPath( ) has 1 (arithmetic addition)

Function 6 - DisplayShortestPath ( ) 1 (Display operation)

Function 7 - FindShortestPath ( ) 4 (Display, Read, Type-Casting & increment operations)

= 3 + 3 + 3 + 1 + 1 + 1 + 4 = 16

FR = Number of functions \* = 10 \* 16 = 160

**3. Non functional Requirement (NFR)**

NFR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 160 + 0 = 160

**5. Product Complexity (PC)**

PC = IOC \* RC = 67 \* 160 = 10720

**6. Personal Complexity Attributes (PCA)**

PCA = 1.00 (Suppose Programmer Capability = Normal)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((10720 \* 1.00) + 0 + 0 + 0 ) \* 1 = 10720

**RBC measure on Bellman Ford Algorithm implemented with C#**

Given a graph G ( V, E)



**Input**

Extracting from the graph above:

N = Number of vertices V = 6, count (N) = 1

V = { S, A, B, C, D, t } , count = 6

M = Number of Edges = 9, count (M) = 1

E = Edges map { 5, -2, 2, 1, 2, 3, 7, 3, 10}, count (E) = 9

Distance = vertex distance map of size |V| which must be initialized to ∞ for all vertices u

in the graph, except for source vertex S, distance(s) = 0

Count (distance) = N = 6

Predecessor = A predecessor map (option 4) of size N, which must be initialized to Null count (distance) = N = 6

s = Source vertex, count(s) = 1

t = Target vertex, count(t) = 1

No of input = count (N) + count (V) + count (M) + count (E) + count (Distance) + count (predecessor) + count (S) + count (t)

= 1 + 6 + 1 + 9 + 6 + 6 + 1 + 1 = 31

**Output**

* Distance = |V| array of the computed shortest distance from the Source to every vertex in the graph

Count (Distance) = 6

* The shortest path constructed from a given source s to the target t

Path = { S, E, A, B, t }, Count (path) = 5

* The cost/weight of the shortest path found, computed as;

costs = -2 + 2 + 1 + 3 = 4, count (cost ) = 1

* No of output = count (Distance) + count (Path) + count (cost)

= 6 + 5 + 1 = 12

* No of interface = 1 (console window)
* No of files = 1 (Data storage)

1. **Input Output Complexity (IOC)**

IOC = No of input + No of output + No of Interface + No of files

= 31 + 12 + 1 + 1 = 45

1. **Functional Requirement (FR)**

No of functions = (BellManFordShortestPath ( ), SetNoOfNodes ( ), GetNoOfNodes ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), FindShortestPath ( )) = 8

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- processes such as Arithmetic, Computation, Display, etc.

Function 1 - ShortestPath ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 2 - ReadNodes ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 3 - ReadWeight ( ) has 3 (Display, Read, Type-casting operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - FindShortestPath ( ) 2 (Display, arithmetic addition)

= 3 + 3 + 3 + 1 + 2 = 12

FR = Number of functions \* = 8 \* 12 = 96

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 96 + 0 = 96

**5. Product Complexity (PC)**

PC = IOC \* RC = 45 \* 96 = 4320

**6**. **Personal Complexity Attributes (PCA)**

PCA = 1.17 (Suppose Programmer Capability = Low)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((4320 \* 1.17) + 0 + 0 + 0 ) \* 1 = 5054.4

**RBC measure on Bellman Ford Algorithm implemented with C++**

1. **Input Output Complexity (IOC)**

No of input = count (N) + count (V) + count (M) + count (E) + count (Distance) + count (predecessor) + count (S) + count (t)

= 1 + 6 + 1 + 9 + 6 + 6 + 1 + 1 = 31

* No of output = count (Distance) + count (Path) + count (cost)

= 6 + 5 + 1 = 12

* No of interface = 1 (console window)
* No of files = 1 (Data storage)

IOC = No of input + No of output + No of Interface + No of files

= 31 + 12 + 1 + 1 = 45

1. **Functional Requirement (FR)**

No of functions = (BellManFordShortestPath ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( )) = 5

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- processes such as Arithmetic, Computation, Display, etc.

Function 1 - ShortestPath ( ) has 2 (Display operations, Read operations)

Function 2 - ReadNodes ( ) has 2 (Display operations, Read operations)

Function 3 - ReadWeight ( ) has 2 (Display, Read operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - BellmanFordShortestPath ( ) has 2 (Display, arithmetic addition)

= 2 + 2 + 2 + 1 + 2 = 9

FR = Number of functions \* = 5 \* 9 = 45

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 45 + 0 = 45

**5. Product Complexity (PC)**

PC = IOC \* RC = 45 \* 45 = 2025

**6**. **Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((2025 \* 0.90) + 0 + 0 + 0 ) \* 1 = 1822.5

**RBC measure on Bellman Ford Algorithm implemented with VB 6.0**

1. **Input Output Complexity (IOC)**

No of input = count (N) + count (V) + count (M) + count (E) + count (Distance) + count (predecessor) + count (S) + count (t)

= 1 + 6 + 1 + 9 + 6 + 6 + 1 + 1 = 31

* No of output = count (Distance) + count (Path) + count (cost)

= 6 + 5 + 1 = 12

* No of interface = 1 (console window)
* No of files = 1 (Data storage)

IOC = No of input + No of output + No of Interface + No of files

= 31 + 12 + 1 + 1 = 45

1. **Functional Requirement (FR)**

No of functions = (FindShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( )) = 4

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- processes such as Arithmetic, Computation, Display, etc.

Function 1 - ReadNodes ( ) has 2 (Display operations, Read operations)

Function 2 - ReadWeight ( ) has 2 (Display, Read operations)

Function 3 - DisplayWeight ( ) has 1 (Display operations)

Function 4 - BellmanFordShortestPath ( ) has 2 (Display, arithmetic addition)

= 2 + 2 + 1 + 2 = 7

FR = Number of functions \* = 4 \* 7 = 28

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 28 + 0 = 28

**5. Product Complexity (PC)**

PC = IOC \* RC = 45 \* 28 = 1260

**6**. **Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((1260 \* 0.90) + 0 + 0 + 0 ) \* 1 = 1134

**RBC measure on Bellman Ford Algorithm implemented with JAVA**

1. **Input Output Complexity (IOC)**

No of input = count (N) + count (V) + count (M) + count (E) + count (Distance) + count (predecessor) + count (S) + count (t)

= 1 + 6 + 1 + 9 + 6 + 6 + 1 + 1 = 31

* No of output = count (Distance) + count (Path) + count (cost)

= 6 + 5 + 1 = 12

* No of interface = 1 (console window)
* No of files = 1 (Data storage)

IOC = No of input + No of output + No of Interface + No of files

= 31 + 12 + 1 + 1 = 45

1. **Functional Requirement (FR)**

No of functions = (BellManFordShortestPath ( ), SetNoOfNodes ( ), GetNoOfNodes ( ), ShortestPath ( ), ReadNodes ( ), ReadWeight ( ), DisplayWeight ( ), FindShortestPath ( )) = 8

No of sub-process/sub-functions ⇒ Every function Fi has zero, one or more sub- processes such as Arithmetic, Computation, Display, etc.

Function 1 - ShortestPath ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 2 - ReadNodes ( ) has 3 (Display operations, Read operations, Type-Casting operations)

Function 3 - ReadWeight ( ) has 3 (Display, Read, Type-casting operations)

Function 4 - DisplayWeight ( ) has 1 (Display operations)

Function 5 - FindShortestPath ( ) has 2 (Display, arithmetic addition)

= 3 + 3 + 3 + 1 + 2 = 12

FR = Number of functions \* = 8 \* 12 = 96

**3. Non functional Requirement (NFR)**

Number of Non-FR = 0 (no quality attribute)

**4. Requirement Complexity (RC)**

RC = FR + NFR = 96 + 0 = 96

**5. Product Complexity (PC)**

PC = IOC \* RC = 45 \* 96 = 4320

**6**. **Personal Complexity Attributes (PCA)**

PCA = 0.90 (Suppose Programmer Capability = High)

**7. Design Constraints Imposed (DCI)**

Number of Constraints = 00 (No directives)

DCI = 0

**8. Interface Complexity (IFC)**

IFC = 0 ( no External Interface required)

**9. Users/Location Complexity (ULC)**

Number of user = 1

Number of location = 1

ULC = No of user \* No of location = 1 \* 1 = 1

**10. System Feature complexity**

SFC = 0 (no specific features required)

**Requirement Based Complexity (RBC)**

RBC = ((PC \* PCA) + DCI + IFC + SFC) \* ULC

= ((4320 \* 0.90) + 0 + 0 + 0 ) \* 1 = 3888