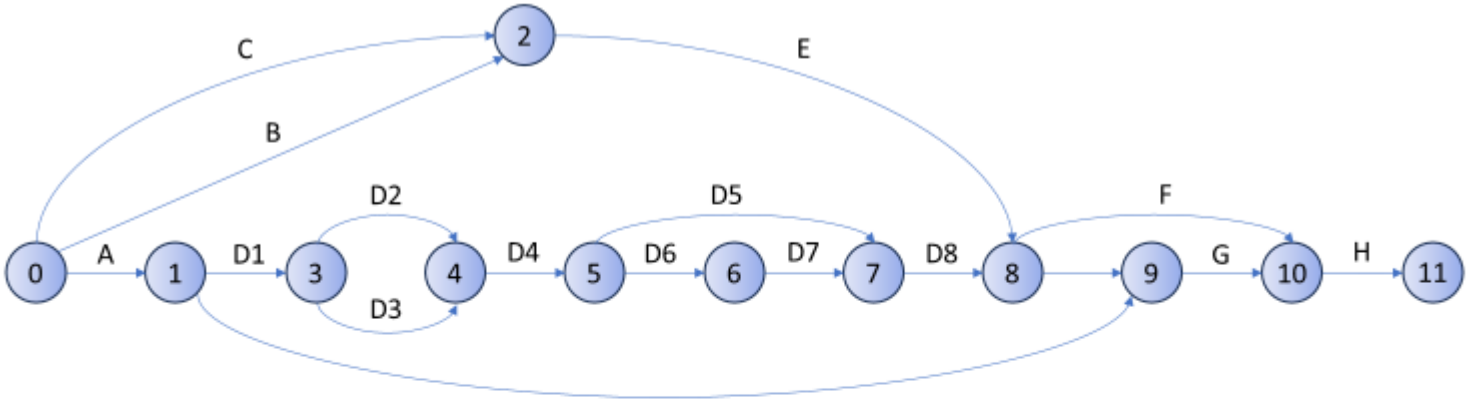
The first set of bestCase, expected and worstCase hours:

taskID	task	predecessorTaskIDs	bestCaseHours	expectedHours	worstCaseHours
A	Describe product		1	4	7
B	Develop marketing strategy		3	6	10
C	Design brochure		4	8	10
D	Develop product prototype				
D1	Requirements analysis	A	3	8	12
D2	Software design	D1	4	10	14
D3	System design	D1	10	14	17
D4	Coding	D2, D3	8	12	19
D5	Write documentation	D4	4	8	19
D6	Unit testing	D4	3	7	17
D7	System testing	D6	2	5	6
D8	Package deliverables	D5, D7	5	6	7
E	Survey potential market	B, C	4	6	8
F	Develop pricing plan	D8, E	3	7	10
G	Develop implementation plan	A, D8	2	5	7
H	Write client proposal	F, G	12	15	19

The second set of bestCase, expected and worstCase hours:

taskID	task	predecessorTaskIDs	BestCase	Expected	WorstCase
A	Describe product		20	30	40
B	Develop marketing strategy		800	1600	2400
C	Design brochure	A	10	15	20
D	Develop product prototype				
D1	Requirements analysis	A	80	120	160
D2	Software design	D1	40	80	120
D3	System design	D1	80	120	160
D4	Coding	D2, D3	120	300	480
D5	Write documentation	D4	40	60	80
D6	Unit testing	D4	24	72	120
D7	System testing	D6	120	220	320
D8	Package deliverables	D5, D7	24	72	120
E	Survey potential market	B, C	24	52	80
F	Develop pricing plan	D8, E	24	52	80
G	Develop implementation plan	A, D8	24	32	40
H	Write client proposal	F, G	24	32	40



A	t0	t1	
B	t0	t2	
C	t0	t2	
D			
D1	t1	t3	
D2	t3	t4	
D3	t3	t4	
D4	t4	t5	
D5	t5	t7	
D6	t5	t6	
D7	t6	t7	
D8	t7	t8	
E	t2	t8	
F	t8	t10	
G	t9	t10	
H	t10	Z	
0	t1	t9	
0	t8	t9	

In [1]:
```
import pulp

# BestCaseHours used
# Create a Linear Programming problem with the goal of minimization
problem = pulp.LpProblem("Minimize Z", pulp.LpMinimize)

# Define variables t0 to Z as continuous variables
t0 = pulp.LpVariable("t0", lowBound=0, cat='Continuous')
t1 = pulp.LpVariable("t1", lowBound=0, cat='Continuous')
t2 = pulp.LpVariable("t2", lowBound=0, cat='Continuous')
t3 = pulp.LpVariable("t3", lowBound=0, cat='Continuous')
t4 = pulp.LpVariable("t4", lowBound=0, cat='Continuous')
t5 = pulp.LpVariable("t5", lowBound=0, cat='Continuous')
t6 = pulp.LpVariable("t6", lowBound=0, cat='Continuous')
t7 = pulp.LpVariable("t7", lowBound=0, cat='Continuous')
```

```
t8 = pulp.LpVariable("t8", lowBound=0, cat='Continuous')
t9 = pulp.LpVariable("t9", lowBound=0, cat='Continuous')
t10 = pulp.LpVariable("t10", lowBound=0, cat='Continuous')
Z = pulp.LpVariable("Z", lowBound=0, cat='Continuous')

# Define the constraints
problem += t1 - t0 >= 1
problem += t2 - t0 >= 3
problem += t2 - t0 >= 4
problem += t3 - t1 >= 3
problem += t4 - t3 >= 4
problem += t4 - t3 >= 10
problem += t5 - t4 >= 8
problem += t7 - t5 >= 4
problem += t6 - t5 >= 3
problem += t7 - t6 >= 2
problem += t8 - t7 >= 5
problem += t8 - t2 >= 4
problem += t10 - t8 >= 3
problem += t10 - t9 >= 2
problem += Z - t10 >= 12
problem += t9 - t1 >= 0
problem += t9 - t8 >= 0

# Define the objective function to minimize Z
problem += Z

# Solve the problem
problem.solve()

# Print the results
print("BestCaseStatus:", pulp.LpStatus[problem.status])
for var in problem.variables():
    print(var.name, "=", var.varValue)
print("Z =", pulp.value(problem.objective))
```

```
BestCaseStatus: Optimal
Z = 47.0
t0 = 0.0
t1 = 1.0
t10 = 35.0
t2 = 28.0
t3 = 4.0
t4 = 14.0
t5 = 22.0
t6 = 25.0
t7 = 27.0
t8 = 32.0
t9 = 33.0
Z = 47.0
```

```
C:\Users\kshen\AppData\Roaming\Python\Python39\site-packages\pulp\pulp.py:1352: UserWarning: Spaces are not permitted i
n the name. Converted to '_'
  warnings.warn("Spaces are not permitted in the name. Converted to '_'")
```

In [2]:

```python
# ExpectedHours used
# Create a Linear Programming problem with the goal of minimization
problem = pulp.LpProblem("Minimize Z", pulp.LpMinimize)

# Define variables t0 to Z as continuous variables
t0 = pulp.LpVariable("t0", lowBound=0, cat='Continuous')
t1 = pulp.LpVariable("t1", lowBound=0, cat='Continuous')
t2 = pulp.LpVariable("t2", lowBound=0, cat='Continuous')
t3 = pulp.LpVariable("t3", lowBound=0, cat='Continuous')
t4 = pulp.LpVariable("t4", lowBound=0, cat='Continuous')
t5 = pulp.LpVariable("t5", lowBound=0, cat='Continuous')
t6 = pulp.LpVariable("t6", lowBound=0, cat='Continuous')
t7 = pulp.LpVariable("t7", lowBound=0, cat='Continuous')
t8 = pulp.LpVariable("t8", lowBound=0, cat='Continuous')
t9 = pulp.LpVariable("t9", lowBound=0, cat='Continuous')
t10 = pulp.LpVariable("t10", lowBound=0, cat='Continuous')
Z = pulp.LpVariable("Z", lowBound=0, cat='Continuous')

# Define the constraints
problem += t1 - t0 >= 4
problem += t2 - t0 >= 6
problem += t2 - t0 >= 8
problem += t3 - t1 >= 8
problem += t4 - t3 >= 10
problem += t4 - t3 >= 14
problem += t5 - t4 >= 12
problem += t7 - t5 >= 8
```

```
problem += t6 - t5 >= 7
problem += t7 - t6 >= 5
problem += t8 - t7 >= 6
problem += t8 - t2 >= 6
problem += t10 - t8 >= 7
problem += t10 - t9 >= 5
problem += Z - t10 >= 15
problem += t9 - t1 >= 0
problem += t9 - t8 >= 0

# Define the objective function to minimize Z
problem += Z

# Solve the problem
problem.solve()

# Print the results
print("ExpectedStatus:", pulp.LpStatus[problem.status])
for var in problem.variables():
    print(var.name, "=", var.varValue)
print("Z =", pulp.value(problem.objective))
```

```
ExpectedStatus: Optimal
Z = 78.0
t0 = 0.0
t1 = 4.0
t10 = 63.0
t2 = 50.0
t3 = 12.0
t4 = 26.0
t5 = 38.0
t6 = 45.0
t7 = 50.0
t8 = 56.0
t9 = 58.0
Z = 78.0
```

In [3]:

```
# WorstCaseHours used
# Create a Linear Programming problem with the goal of minimization
problem = pulp.LpProblem("Minimize Z", pulp.LpMinimize)

# Define variables t0 to Z as continuous variables
t0 = pulp.LpVariable("t0", lowBound=0, cat='Continuous')
t1 = pulp.LpVariable("t1", lowBound=0, cat='Continuous')
t2 = pulp.LpVariable("t2", lowBound=0, cat='Continuous')
```

```
t3 = pulp.LpVariable("t3", lowBound=0, cat='Continuous')
t4 = pulp.LpVariable("t4", lowBound=0, cat='Continuous')
t5 = pulp.LpVariable("t5", lowBound=0, cat='Continuous')
t6 = pulp.LpVariable("t6", lowBound=0, cat='Continuous')
t7 = pulp.LpVariable("t7", lowBound=0, cat='Continuous')
t8 = pulp.LpVariable("t8", lowBound=0, cat='Continuous')
t9 = pulp.LpVariable("t9", lowBound=0, cat='Continuous')
t10 = pulp.LpVariable("t10", lowBound=0, cat='Continuous')
Z = pulp.LpVariable("Z", lowBound=0, cat='Continuous')

# Define the constraints
problem += t1 - t0 >= 7
problem += t2 - t0 >= 10
problem += t2 - t0 >= 10
problem += t3 - t1 >= 12
problem += t4 - t3 >= 14
problem += t4 - t3 >= 17
problem += t5 - t4 >= 19
problem += t7 - t5 >= 19
problem += t6 - t5 >= 17
problem += t7 - t6 >= 6
problem += t8 - t7 >= 7
problem += t8 - t2 >= 8
problem += t10 - t8 >= 10
problem += t10 - t9 >= 7
problem += Z - t10 >= 19
problem += t9 - t1 >= 0
problem += t9 - t8 >= 0

# Define the objective function to minimize Z
problem += Z

# Solve the problem
problem.solve()

# Print the results
print("WorstCaseStatus:", pulp.LpStatus[problem.status])
for var in problem.variables():
    print(var.name, "=", var.varValue)
print("Z =", pulp.value(problem.objective))
```

```
WorstCaseStatus: Optimal
Z = 114.0
t0 = 0.0
t1 = 7.0
t10 = 95.0
t2 = 77.0
t3 = 19.0
t4 = 36.0
t5 = 55.0
t6 = 72.0
t7 = 78.0
t8 = 85.0
t9 = 88.0
Z = 114.0
```

In [4]:

```python
import pulp

# The 2nd set of BestCaseHours used
# Create a Linear Programming problem with the goal of minimization
problem = pulp.LpProblem("Minimize Z", pulp.LpMinimize)

# Define variables t0 to Z as continuous variables
t0 = pulp.LpVariable("t0", lowBound=0, cat='Continuous')
t1 = pulp.LpVariable("t1", lowBound=0, cat='Continuous')
t2 = pulp.LpVariable("t2", lowBound=0, cat='Continuous')
t3 = pulp.LpVariable("t3", lowBound=0, cat='Continuous')
t4 = pulp.LpVariable("t4", lowBound=0, cat='Continuous')
t5 = pulp.LpVariable("t5", lowBound=0, cat='Continuous')
t6 = pulp.LpVariable("t6", lowBound=0, cat='Continuous')
t7 = pulp.LpVariable("t7", lowBound=0, cat='Continuous')
t8 = pulp.LpVariable("t8", lowBound=0, cat='Continuous')
t9 = pulp.LpVariable("t9", lowBound=0, cat='Continuous')
t10 = pulp.LpVariable("t10", lowBound=0, cat='Continuous')
Z = pulp.LpVariable("Z", lowBound=0, cat='Continuous')

# Define the constraints
problem += t1 - t0 >= 20
problem += t2 - t0 >= 800
problem += t2 - t0 >= 10
problem += t3 - t1 >= 80
problem += t4 - t3 >= 40
problem += t4 - t3 >= 80
problem += t5 - t4 >= 120
problem += t7 - t5 >= 40
problem += t6 - t5 >= 24
```

```
problem += t7 - t6 >= 120
problem += t8 - t7 >= 24
problem += t8 - t2 >= 24
problem += t10 - t8 >= 24
problem += t10 - t9 >= 24
problem += Z - t10 >= 24
problem += t9 - t1 >= 0
problem += t9 - t8 >= 0

# Define the objective function to minimize Z
problem += Z

# Solve the problem
problem.solve()

# Print the results
print("Set#2 BestCaseStatus:", pulp.LpStatus[problem.status])
for var in problem.variables():
    print(var.name, "=", var.varValue)
print("Z =", pulp.value(problem.objective))
```

```
Set#2 BestCaseStatus: Optimal
Z = 872.0
t0 = 0.0
t1 = 20.0
t10 = 848.0
t2 = 800.0
t3 = 100.0
t4 = 180.0
t5 = 300.0
t6 = 324.0
t7 = 444.0
t8 = 824.0
t9 = 824.0
Z = 872.0
```

In [5]:

```
# The 2nd set of ExpectedHours used
# Create a Linear Programming problem with the goal of minimization
problem = pulp.LpProblem("Minimize Z", pulp.LpMinimize)

# Define variables t0 to Z as continuous variables
t0 = pulp.LpVariable("t0", lowBound=0, cat='Continuous')
t1 = pulp.LpVariable("t1", lowBound=0, cat='Continuous')
t2 = pulp.LpVariable("t2", lowBound=0, cat='Continuous')
t3 = pulp.LpVariable("t3", lowBound=0, cat='Continuous')
```

```
t4 = pulp.LpVariable("t4", lowBound=0, cat='Continuous')
t5 = pulp.LpVariable("t5", lowBound=0, cat='Continuous')
t6 = pulp.LpVariable("t6", lowBound=0, cat='Continuous')
t7 = pulp.LpVariable("t7", lowBound=0, cat='Continuous')
t8 = pulp.LpVariable("t8", lowBound=0, cat='Continuous')
t9 = pulp.LpVariable("t9", lowBound=0, cat='Continuous')
t10 = pulp.LpVariable("t10", lowBound=0, cat='Continuous')
Z = pulp.LpVariable("Z", lowBound=0, cat='Continuous')

# Define the constraints
problem += t1 - t0 >= 30
problem += t2 - t0 >= 1600
problem += t2 - t0 >= 15
problem += t3 - t1 >= 120
problem += t4 - t3 >= 80
problem += t4 - t3 >= 120
problem += t5 - t4 >= 300
problem += t7 - t5 >= 60
problem += t6 - t5 >= 72
problem += t7 - t6 >= 220
problem += t8 - t7 >= 72
problem += t8 - t2 >= 52
problem += t10 - t8 >= 52
problem += t10 - t9 >= 52
problem += Z - t10 >= 52
problem += t9 - t1 >= 0
problem += t9 - t8 >= 0

# Define the objective function to minimize Z
problem += Z

# Solve the problem
problem.solve()

# Print the results
print("Set#2 ExpectedStatus:", pulp.LpStatus[problem.status])
for var in problem.variables():
    print(var.name, "=", var.varValue)
print("Z =", pulp.value(problem.objective))
```

```
Set#2 ExpectedStatus: Optimal
Z = 1756.0
t0 = 0.0
t1 = 30.0
t10 = 1704.0
t2 = 1600.0
t3 = 150.0
t4 = 270.0
t5 = 570.0
t6 = 642.0
t7 = 862.0
t8 = 1652.0
t9 = 1652.0
Z = 1756.0
```

In [6]:

```python
# The 2nd set of WorstCaseHours used
# Create a Linear Programming problem with the goal of minimization
problem = pulp.LpProblem("Minimize Z", pulp.LpMinimize)

# Define variables t0 to Z as continuous variables
t0 = pulp.LpVariable("t0", lowBound=0, cat='Continuous')
t1 = pulp.LpVariable("t1", lowBound=0, cat='Continuous')
t2 = pulp.LpVariable("t2", lowBound=0, cat='Continuous')
t3 = pulp.LpVariable("t3", lowBound=0, cat='Continuous')
t4 = pulp.LpVariable("t4", lowBound=0, cat='Continuous')
t5 = pulp.LpVariable("t5", lowBound=0, cat='Continuous')
t6 = pulp.LpVariable("t6", lowBound=0, cat='Continuous')
t7 = pulp.LpVariable("t7", lowBound=0, cat='Continuous')
t8 = pulp.LpVariable("t8", lowBound=0, cat='Continuous')
t9 = pulp.LpVariable("t9", lowBound=0, cat='Continuous')
t10 = pulp.LpVariable("t10", lowBound=0, cat='Continuous')
Z = pulp.LpVariable("Z", lowBound=0, cat='Continuous')

# Define the constraints
problem += t1 - t0 >= 40
problem += t2 - t0 >= 2400
problem += t2 - t0 >= 20
problem += t3 - t1 >= 160
problem += t4 - t3 >= 120
problem += t4 - t3 >= 160
problem += t5 - t4 >= 480
problem += t7 - t5 >= 80
problem += t6 - t5 >= 120
problem += t7 - t6 >= 320
problem += t8 - t7 >= 120
```

```
problem += t8 - t2 >= 80
problem += t10 - t8 >= 80
problem += t10 - t9 >= 40
problem += Z - t10 >= 40
problem += t9 - t1 >= 0
problem += t9 - t8 >= 0

# Define the objective function to minimize Z
problem += Z

# Solve the problem
problem.solve()

# Print the results
print("Set#2 WorstCaseStatus:", pulp.LpStatus[problem.status])
for var in problem.variables():
    print(var.name, "=", var.varValue)
print("Z =", pulp.value(problem.objective))
```

```
Set#2 WorstCaseStatus: Optimal
Z = 2600.0
t0 = 0.0
t1 = 40.0
t10 = 2560.0
t2 = 2400.0
t3 = 200.0
t4 = 360.0
t5 = 840.0
t6 = 960.0
t7 = 1280.0
t8 = 2480.0
t9 = 2520.0
Z = 2600.0
```

In [ ]: