```
──────────────── MODULE WorkflowRepairTest ────────────────
LOCAL INSTANCE TLC

RepairTests ≜

should not repair trivial, valid workflow

⟨LET Repair ≜ INSTANCE WorkflowRepair WITH
  Connections ←
  {
  },
  Tasks ←
  {  [name ↦ "EVI", repeatable ↦ FALSE, group ↦ "both"]
  },
  Workflow ← ⟨"EVI"⟩,
  MaxDepth ← 3

  ACTUAL ≜ Repair!Recommendation

  EXPECTED ≜ ⟨"EVI"⟩
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should not repair trivial, valid workflow")
    )

should repair trivial, invalid workflow

, LET Repair ≜ INSTANCE WorkflowRepair WITH
    Connections ←
    {  [name ↦ "has_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
    },
    Tasks ←
    {  [name ↦ "EVI", repeatable ↦ FALSE, group ↦ "both"]
    ,  [name ↦ "XRAY", repeatable ↦ FALSE, group ↦ "both"]
    },
    Workflow ← ⟨"XRAY", "EVI"⟩,
    MaxDepth ← 3

    ACTUAL ≜ Repair!Recommendation

    EXPECTED ≜ ⟨"EVI", "XRAY"⟩
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should repair trivial, invalid workflow")
    )
```

should repair invalid workflow with missing mandatory tasks I

```
, LET Repair ≜ INSTANCE WorkflowRepair WITH
    Connections ←
    {  [name ↦ “has_mandatory_successor”, srcName ↦ “EVI”, dstName ↦ “XRAY”]
    ,  [name ↦ “has_mandatory_predecessor”, srcName ↦ “XRAY”, dstName ↦ “EVI”]
    ,  [name ↦ “has_mandatory_successor”, srcName ↦ “XRAY”, dstName ↦ “IVI”]
    ,  [name ↦ “has_mandatory_predecessor”, srcName ↦ “IVI”, dstName ↦ “XRAY”]
    },
    Tasks ←
    {  [name ↦ “EVI”, repeatable ↦ TRUE, group ↦ “both”]
    ,  [name ↦ “XRAY”, repeatable ↦ TRUE, group ↦ “both”]
    ,  [name ↦ “IVI”, repeatable ↦ TRUE, group ↦ “both”]
    },
    Workflow ← ⟨“XRAY”, “XRAY”⟩,
    MaxDepth ← 6

    ACTUAL ≜ Repair!Recommendation

    EXPECTED_A ≜
    ⟨ “EVI”, “XRAY”, “IVI”
    ,   “EVI”, “XRAY”, “IVI”
    ⟩
    EXPECTED_B ≜
    ⟨ “EVI”, “XRAY”
    ,   “EVI”
    ,   “IVI”
    ,   “XRAY”, “IVI”
    ⟩
IN
     Assert(
          Print(ACTUAL, ACTUAL = EXPECTED_A ∨ ACTUAL = EXPECTED_B),
          Print(⟨EXPECTED_A, EXPECTED_B⟩, “should repair invalid workflow with missing mandatory tasks I
     )
```

should repair invalid workflow with missing mandatory tasks *II*

```
, LET Repair ≜ INSTANCE WorkflowRepair WITH
    Connections ←
    {  [name ↦ “has_mandatory_successor”, srcName ↦ “EVI”, dstName ↦ “IVI”]

    ,  [name ↦ “has_mandatory_successor”, srcName ↦ “EVI”, dstName ↦ “XRAY”]
    ,  [name ↦ “has_mandatory_predecessor”, srcName ↦ “XRAY”, dstName ↦ “EVI”]
    ,  [name ↦ “has_mandatory_successor”, srcName ↦ “XRAY”, dstName ↦ “IVI”]
    ,  [name ↦ “has_mandatory_predecessor”, srcName ↦ “IVI”, dstName ↦ “XRAY”]
    },
    Tasks ←
    {  [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
    ,   [name ↦ "XRAY", repeatable ↦ TRUE, group ↦ "both"]
    ,   [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow ← ⟨"XRAY", "XRAY"⟩,
    MaxDepth ← 6

    ACTUAL ≜ Repair!Recommendation

    EXPECTED ≜
    ⟨ "EVI", "XRAY", "IVI"
    ,   "EVI", "XRAY", "IVI"
    ⟩
IN
     Assert(
          Print(ACTUAL, ACTUAL = EXPECTED),
          Print(EXPECTED, "should repair invalid workflow with missing mandatory tasks II")
     )
```

should repair invalid workflow with transitive mandatory tasks

```
, LET Repair ≜ INSTANCE WorkflowRepair WITH
    Connections ←
    {  [name ↦ "has_mandatory_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
    ,   [name ↦ "has_mandatory_predecessor", srcName ↦ "XRAY", dstName ↦ "EVI"]

    ,   [name ↦ "has_mandatory_predecessor", srcName ↦ "IVI", dstName ↦ "XRAY"]
    },
    Tasks ←
    {  [name ↦ "EVI", repeatable ↦ TRUE, group ↦ "both"]
    ,   [name ↦ "XRAY", repeatable ↦ TRUE, group ↦ "both"]
    ,   [name ↦ "IVI", repeatable ↦ TRUE, group ↦ "both"]
    },
    Workflow ← ⟨"IVI"⟩,
    MaxDepth ← 3

    ACTUAL ≜ Repair!Recommendation

    EXPECTED ≜ ⟨ "EVI", "XRAY", "IVI"⟩
IN
     Assert(
          Print(ACTUAL, ACTUAL = EXPECTED),
          Print(EXPECTED, "should repair invalid workflow with transitive mandatory tasks")
     )
```

should repair invalid workflow with transitive partial order violations

```
, LET Repair ≜ INSTANCE WorkflowRepair WITH
    Connections ←
    {  [name ↦ "has_successor", srcName ↦ "EVI", dstName ↦ "XRAY"]
    ,  [name ↦ "has_successor", srcName ↦ "XRAY", dstName ↦ "IVI"]
    },
    Tasks ←
    {  [name ↦ "EVI", repeatable ↦ FALSE, group ↦ "both"]
    ,  [name ↦ "XRAY", repeatable ↦ FALSE, group ↦ "both"]
    ,  [name ↦ "IVI", repeatable ↦ FALSE, group ↦ "both"]
    },
    Workflow  ← ⟨"IVI", "EVI"⟩,
    MaxDepth ← 3

    ACTUAL ≜ Repair!Recommendation

    EXPECTED ≜ ⟨ "EVI", "IVI"⟩
IN
    Assert(
        Print(ACTUAL, ACTUAL = EXPECTED),
        Print(EXPECTED, "should repair invalid workflow with transitive partial order violations")
    )

⟩
```