```
from unittest.mock import Mock, create_autospec

import pytest

from swarm_models import OpenAIChat
from swarms.structs import RecursiveWorkflow, Task


def test_add():
    workflow = RecursiveWorkflow(stop_token="<DONE>")
    task = Mock(spec=Task)
    workflow.add(task)
    assert task in workflow.tasks


def test_run():
    workflow = RecursiveWorkflow(stop_token="<DONE>")
    agent1 = create_autospec(OpenAIChat)
    agent2 = create_autospec(OpenAIChat)
    task1 = Task("What's the weather in miami", agent1)
    task2 = Task("What's the weather in miami", agent2)
    workflow.add(task1)
    workflow.add(task2)

    agent1.execute.return_value = "Not done"
    agent2.execute.return_value = "<DONE>"
    workflow.run()

    assert agent1.execute.call_count >= 1
    assert agent2.execute.call_count == 1


def test_run_no_tasks():
    workflow = RecursiveWorkflow(stop_token="<DONE>")
    # No tasks are added to the workflow
    # This should not raise any errors
    workflow.run()


def test_run_stop_token_not_in_result():
    workflow = RecursiveWorkflow(stop_token="<DONE>")
    agent = create_autospec(OpenAIChat)
    task = Task("What's the weather in miami", agent)
    workflow.add(task)

    agent.execute.return_value = "Not done"

    # If the stop token is never found in the result, the workflow could run forever.
    # To prevent this, we'll set a maximum number of iterations.
    max_iterations = 1000
    for _ in range(max_iterations):
        try:
            workflow.run()
        except RecursionError:
            pytest.fail(
                "RecursiveWorkflow.run caused a RecursionError"
            )

    assert agent.execute.call_count == max_iterations



def test_run_stop_token_in_result():
    workflow = RecursiveWorkflow(stop_token="<DONE>")
    agent = create_autospec(OpenAIChat)
    task = Task("What's the weather in miami", agent)
    workflow.add(task)

    agent.execute.return_value = "<DONE>"

    workflow.run()

    # If the stop token is found in the result, the workflow should stop running the task.
    assert agent.execute.call_count == 1
```