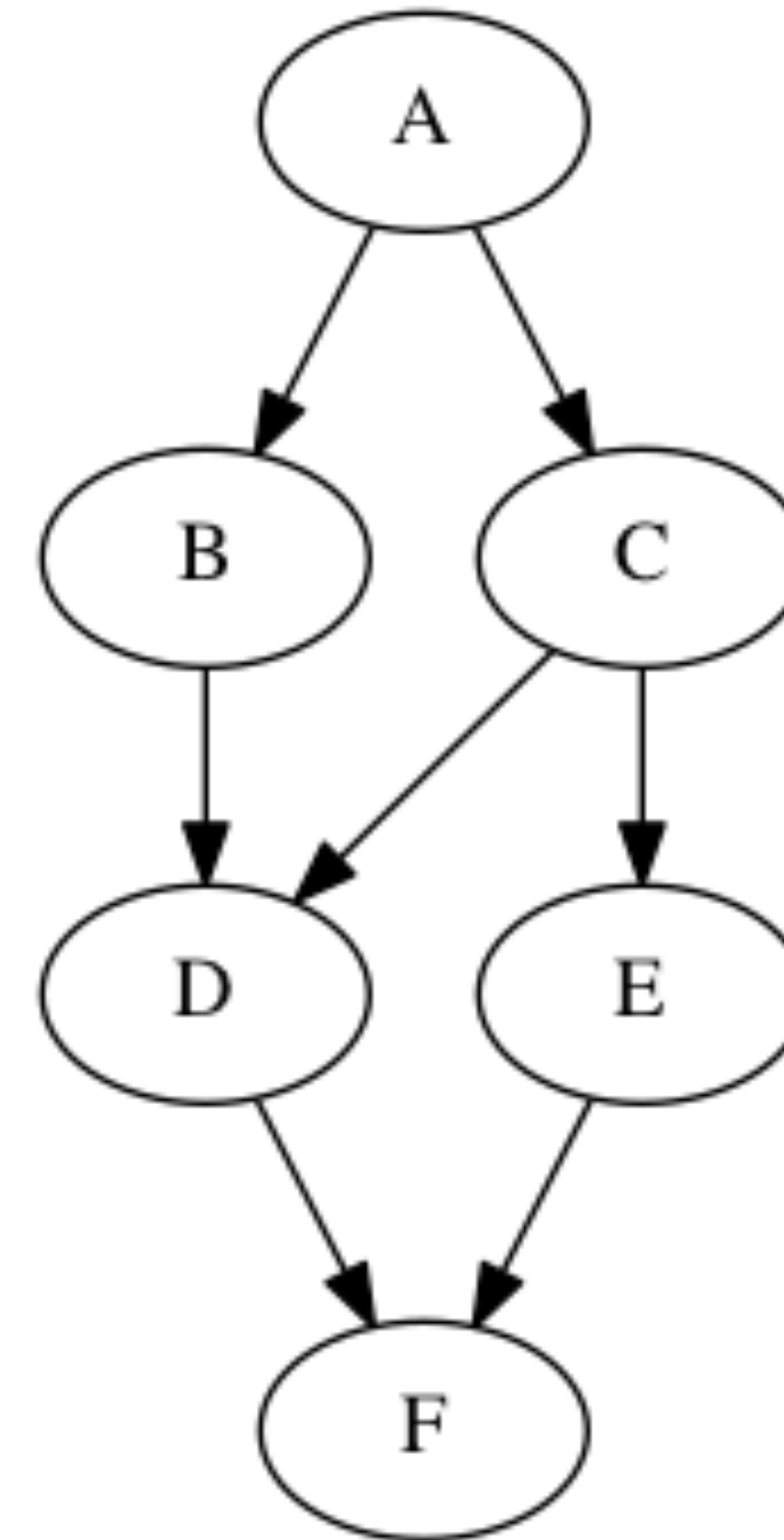


# Worksheet: Computation Graphs for Async-Finish and Future Constructs

1) Can you write pseudocode with async & finish constructs that generates a Computation Graph with the same ordering constraints as the graph on the right? If so, provide a sketch of the program. If not, why not?

2) Can you write pseudocode with future & get constructs that generates a Computation Graph with the same ordering constraints as the graph on the right? If so, provide a sketch of the program. If not, why not?



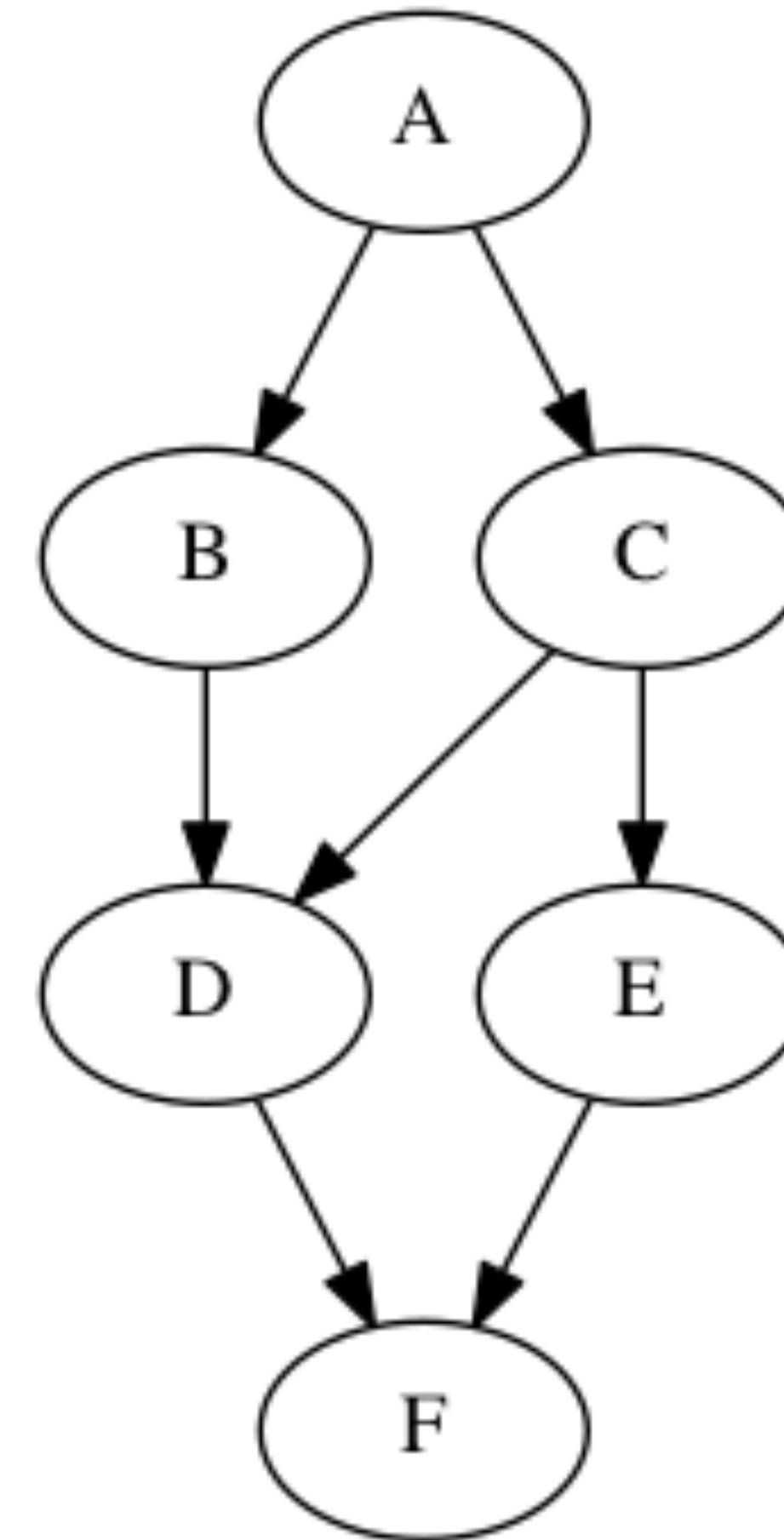
# Worksheet solution

1) Can you write pseudocode with async-finish constructs that generates a Computation Graph with the same ordering constraints as the graph on the right? If so, provide a sketch of the program.

No. Finish cannot be used to ensure that D only waits for B and C, while E waits only for C.

2) Can you write pseudocode with future async-get constructs that generates a Computation Graph with the same ordering constraints as the graph on the right? If so, provide a sketch of the program.

Yes, see program sketch with dummy return values.



## Worksheet solution (contd.)

```
1.  var A = future(() -> {
2.    return "A"; });
3.  var B = future(() -> {
4.    A.get(); return "B"; });
5.  var C = future(() -> {
6.    A.get(); return "C"; });
7.  var D = future(() -> {
8.    // Order of B.get() & C.get() doesn't matter
9.    B.get(); C.get(); return "D"; });
10. var E = future(() -> {
11.   C.get(); return "E"; });
12. var F = future(() -> {
13.   D.get(); E.get(); return "F"; });
14. F.get();
```

