

Worksheet: Cooperative vs Blocking Runtime Schedulers

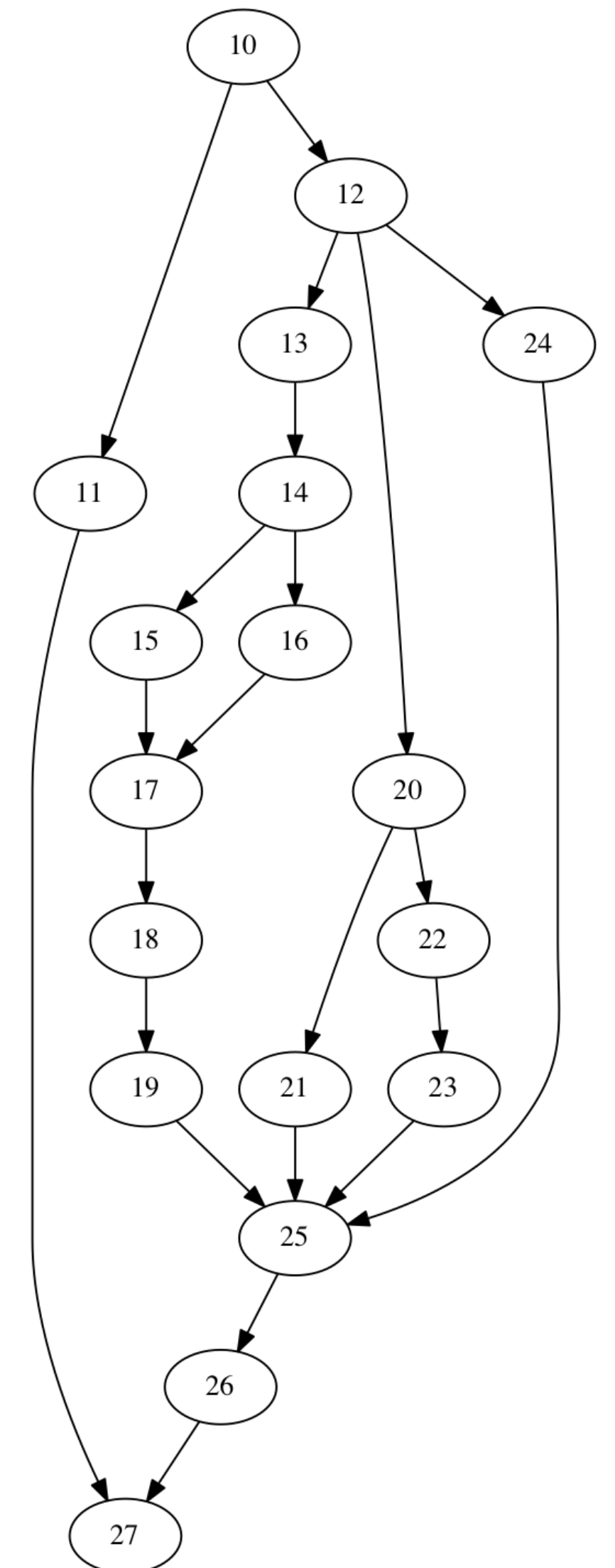
Assume that creating an `async` causes the task to be pushed into the work queue for execution by any available worker thread.

Fill in the following table for the program shown on the right by adding the appropriate number of threads required to execute the program. For the minimum or maximum numbers, your answer must represent a schedule where at some point during the execution all threads are busy executing a task or blocked on some synchronization constraint.

	Minimum number of threads	Maximum number of threads
Cooperative Runtime	1	?
Blocking Runtime	?	?

```

10. finish {
11.   async { S1; }
12.   finish {
13.     async {
14.       finish {
15.         async { S2; }
16.         S3;
17.       }
18.       S4;
19.     }
20.     async {
21.       async { S5; }
22.       S6;
23.     }
24.     S7;
25.   }
26.   S8;
27. }
    
```



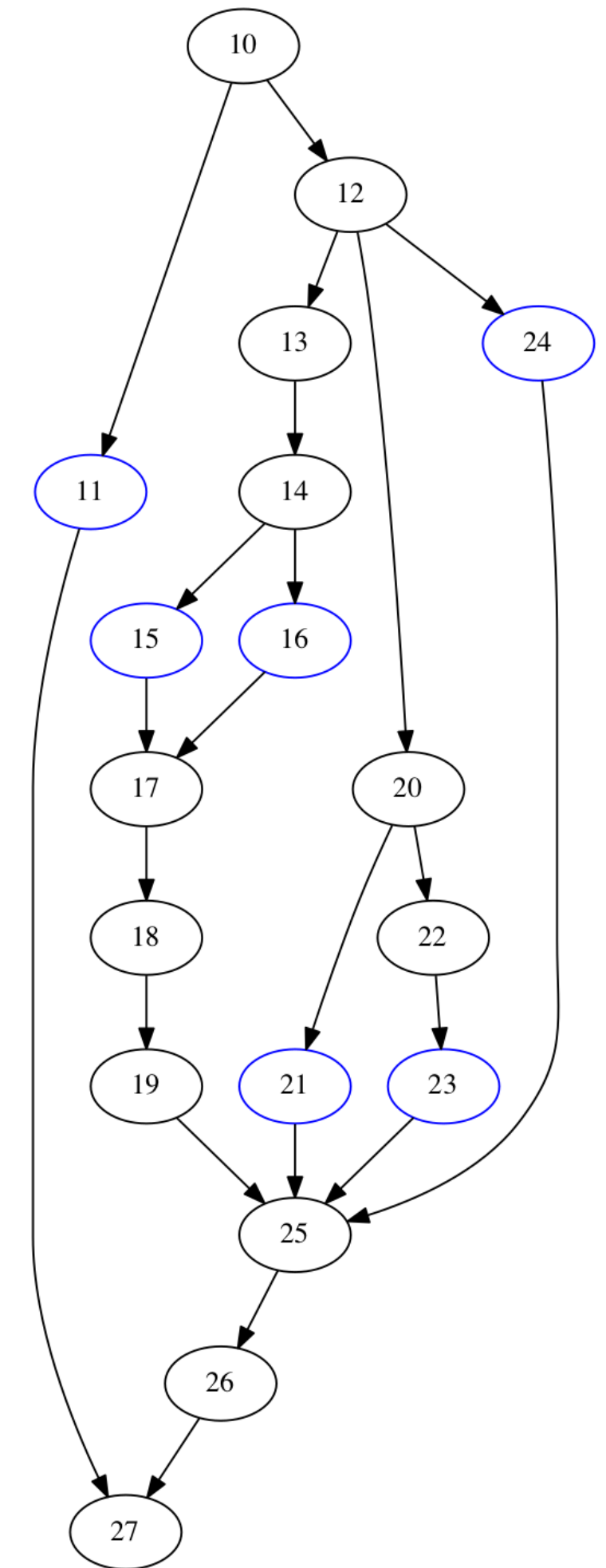
Worksheet: Cooperative vs Blocking Runtime Schedulers

Maximum threads: If we proceed through the graph in top-down manner incrementally, how many maximum leaf nodes can we have?

	Minimum number of threads	Maximum number of threads
Cooperative Runtime	1	6
Blocking Runtime	?	6

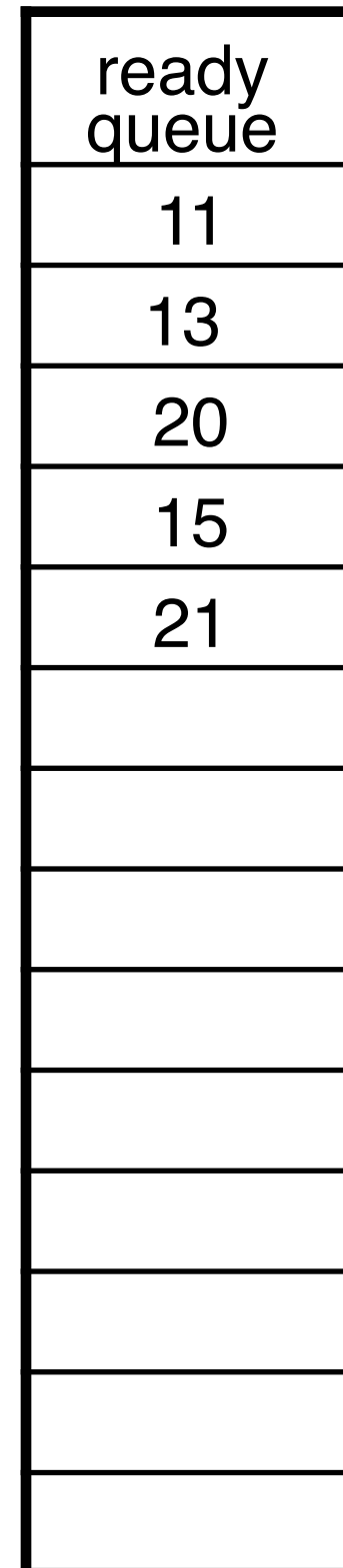
```

10. finish {
11.   async { S1; }
12.   finish {
13.     async {
14.       finish {
15.         async { S2; }
16.         S3;
17.       }
18.       S4;
19.     }
20.     async {
21.       async { S5; }
22.       S6;
23.     }
24.     S7;
25.   }
26.   S8;
27. }
    
```



Worksheet: Cooperative vs Blocking Runtime Schedulers

	Minimum number of threads	Maximum number of threads
Cooperative Runtime	1	6
Blocking Runtime	3	6



```

10. finish {
11.   async { S1; }
12.   finish {
13.     async {
14.       finish {
15.         async { S2; }
16.         S3;
17.       }
18.       S4;
19.     }
20.     async {
21.       async { S5; }
22.       S6;
23.     }
24.     S7;
25.   }
26.   S8;
27. }
    
```

