



Java is a trademark of Sun Microsystems, Inc.

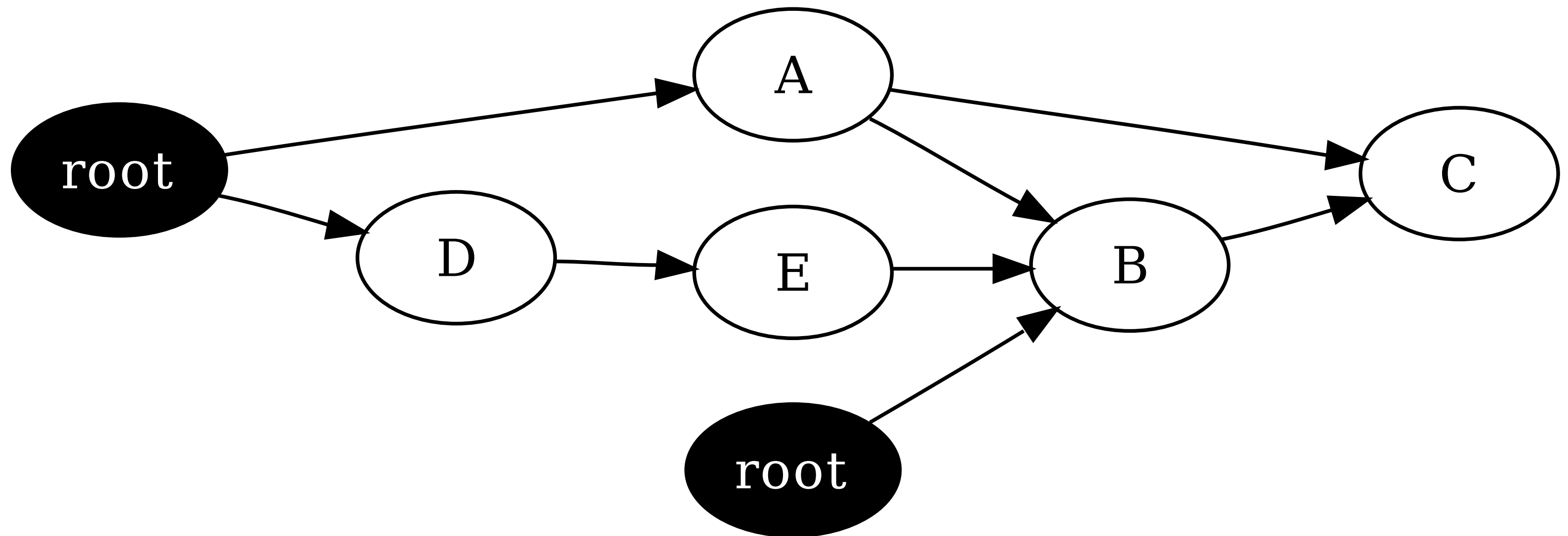


JavaOneSM

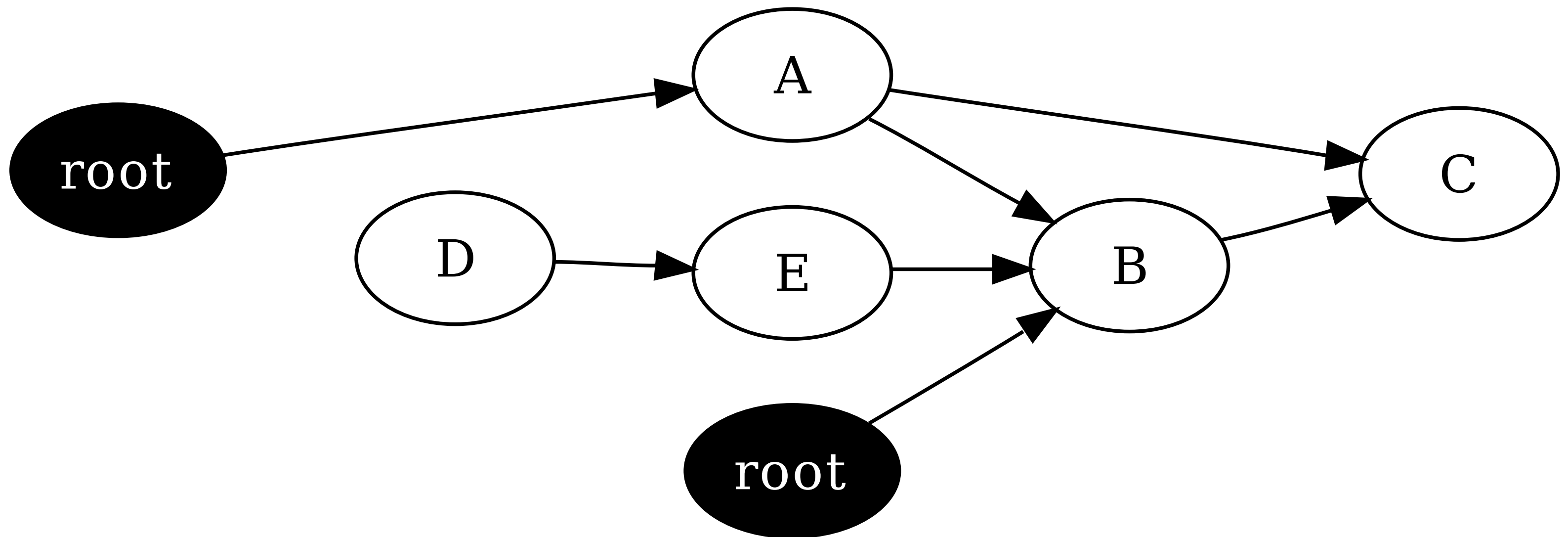
The Ghost in the Virtual Machine A Reference to References

Bob Lee
Google Inc.

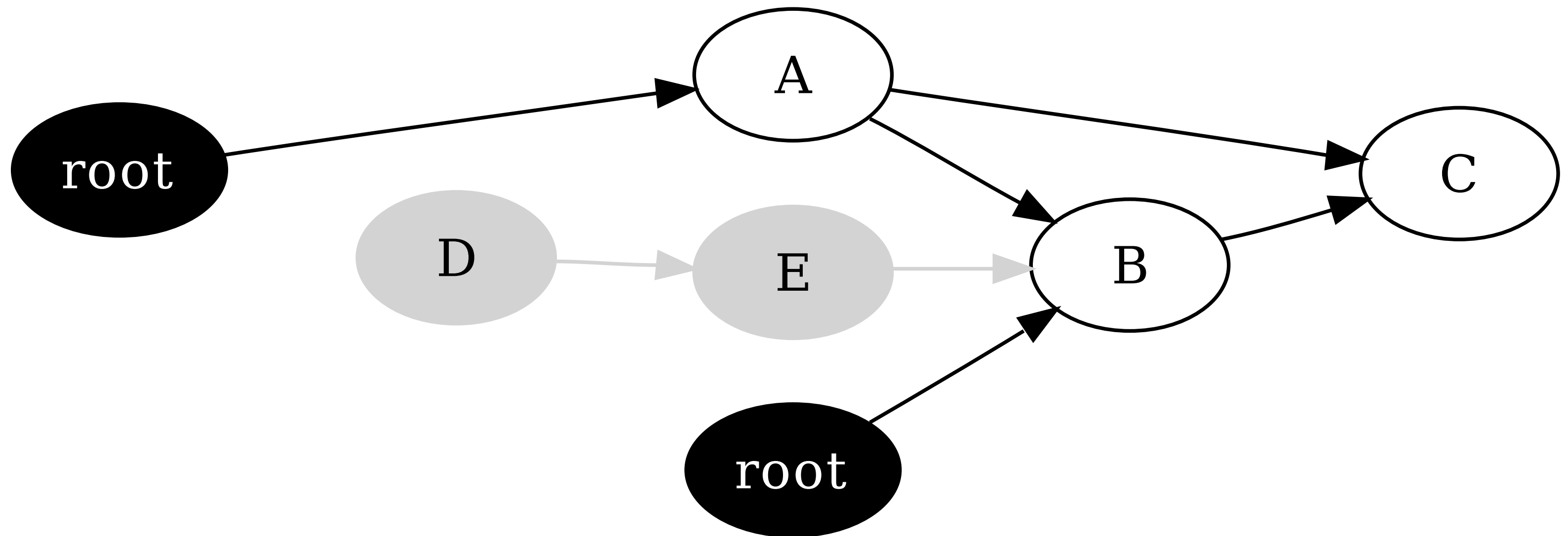
How does garbage collection work?



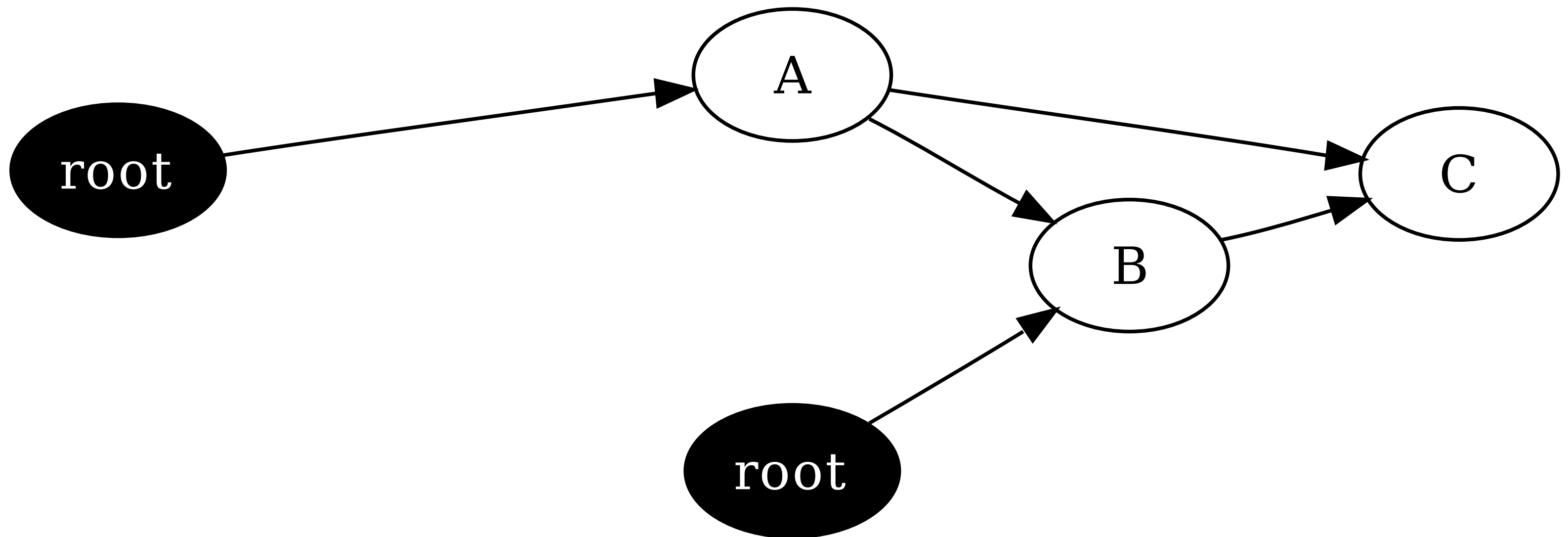
If the reference to D goes away...



We can no longer reach D or E.



The collector reclaims D and E.



Reachability

- > An object is `_reachable_` if a live thread can access it.
- > Examples of heap roots:
 - System classes (which have static state)
 - Thread stacks
 - In-flight exceptions
 - JNI global references
 - The finalizer queue
 - Interned strings
 - etc. (VM-dependent)

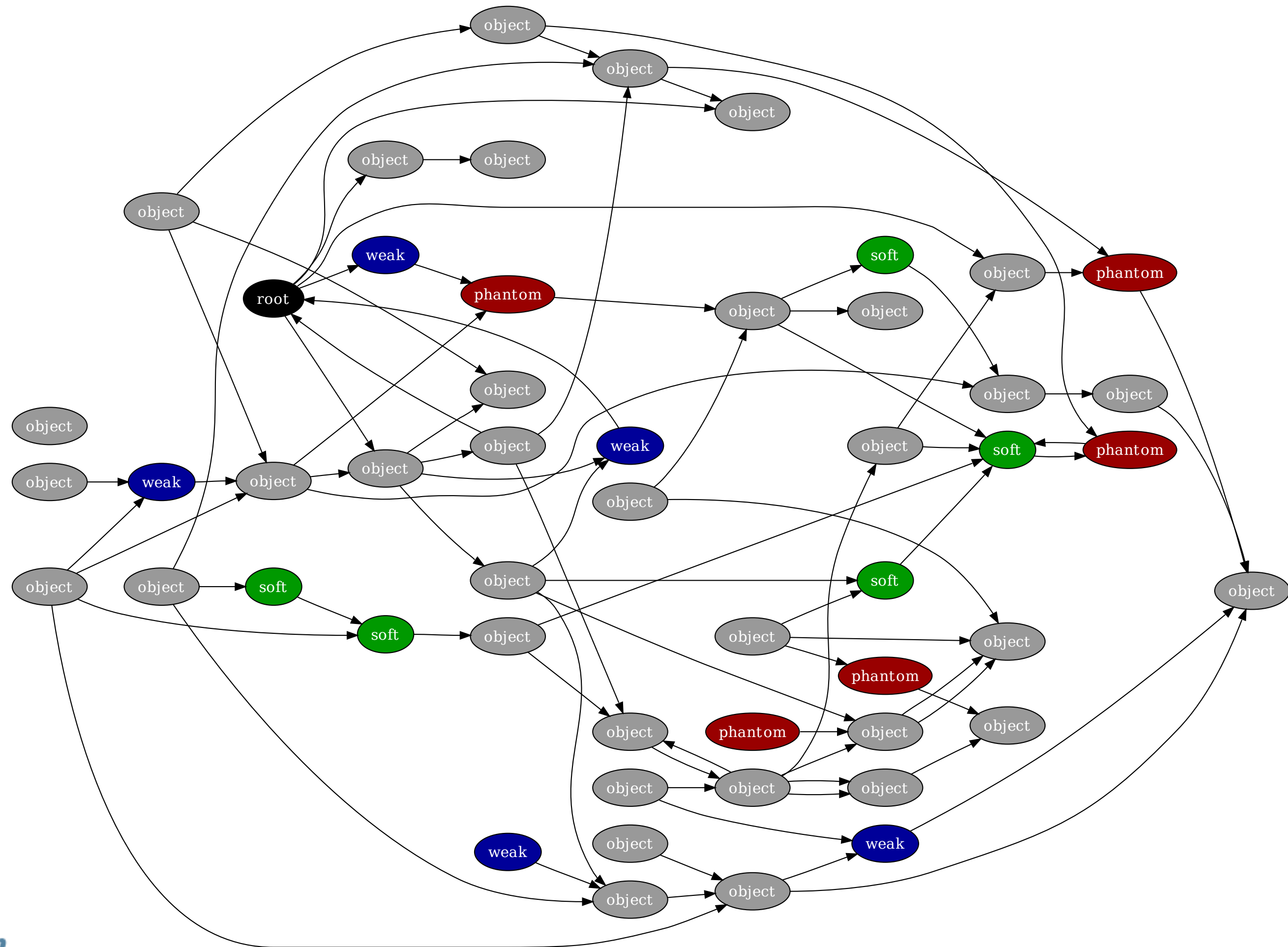
In the beginning, there was the finalizer...

```
public class Foo extends Bar {  
    @Override protected void finalize() throws Throwable {  
        try {  
            ... // Clean up Foo.  
        } finally {  
            super.finalize(); // Clean up Bar.  
        }  
    }  
}
```

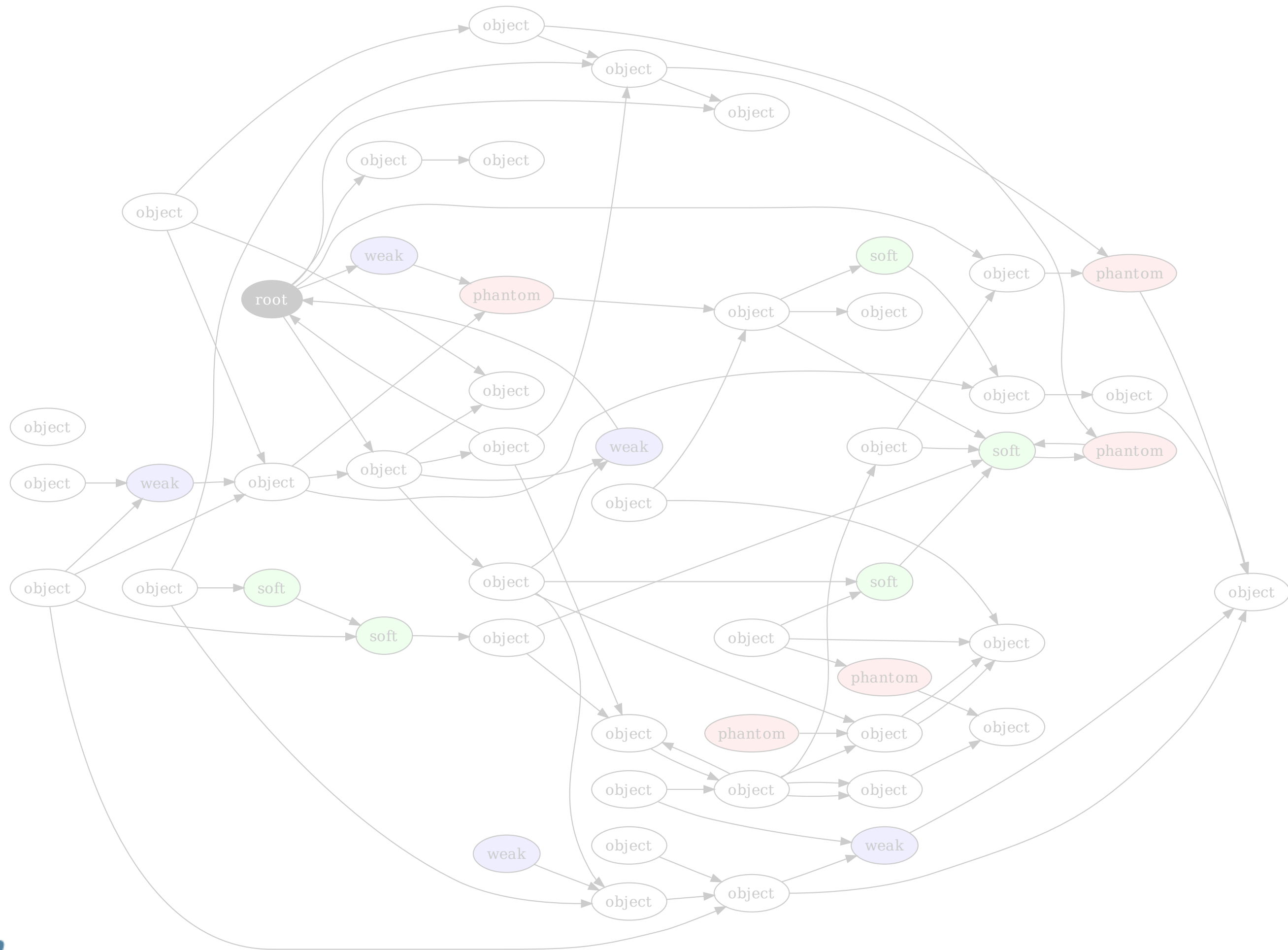
In the beginning, there was the finalizer...

```
public class Foo extends Bar {  
    @Override protected void finalize() throws Throwable {  
        try {  
            ... // Clean up Foo.  
        } finally {  
            super.finalize(); // Clean up Bar.  
        }  
    }  
}
```


A mark-and-sweep collector cycle



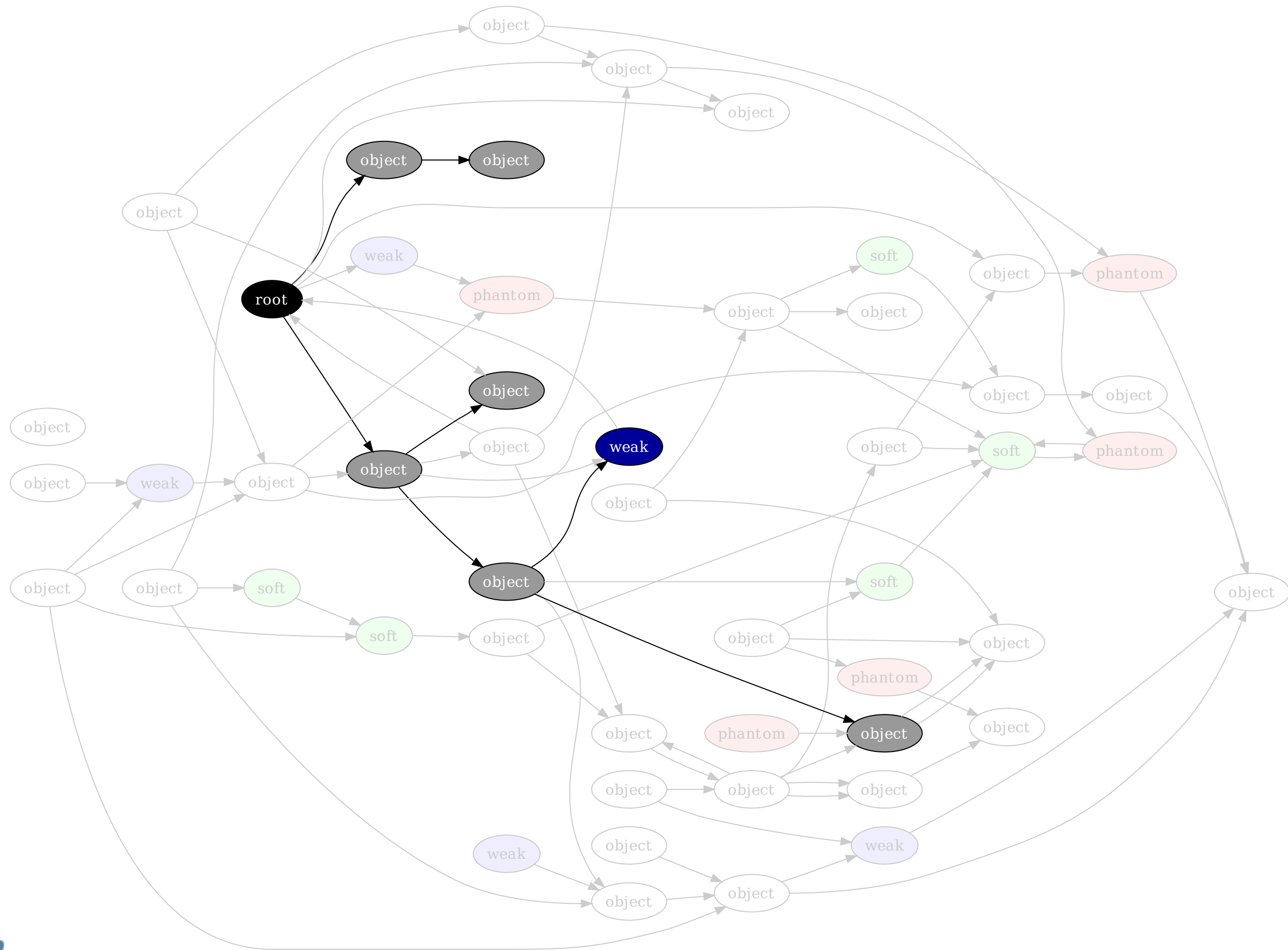
1. Nothing is marked, to start.



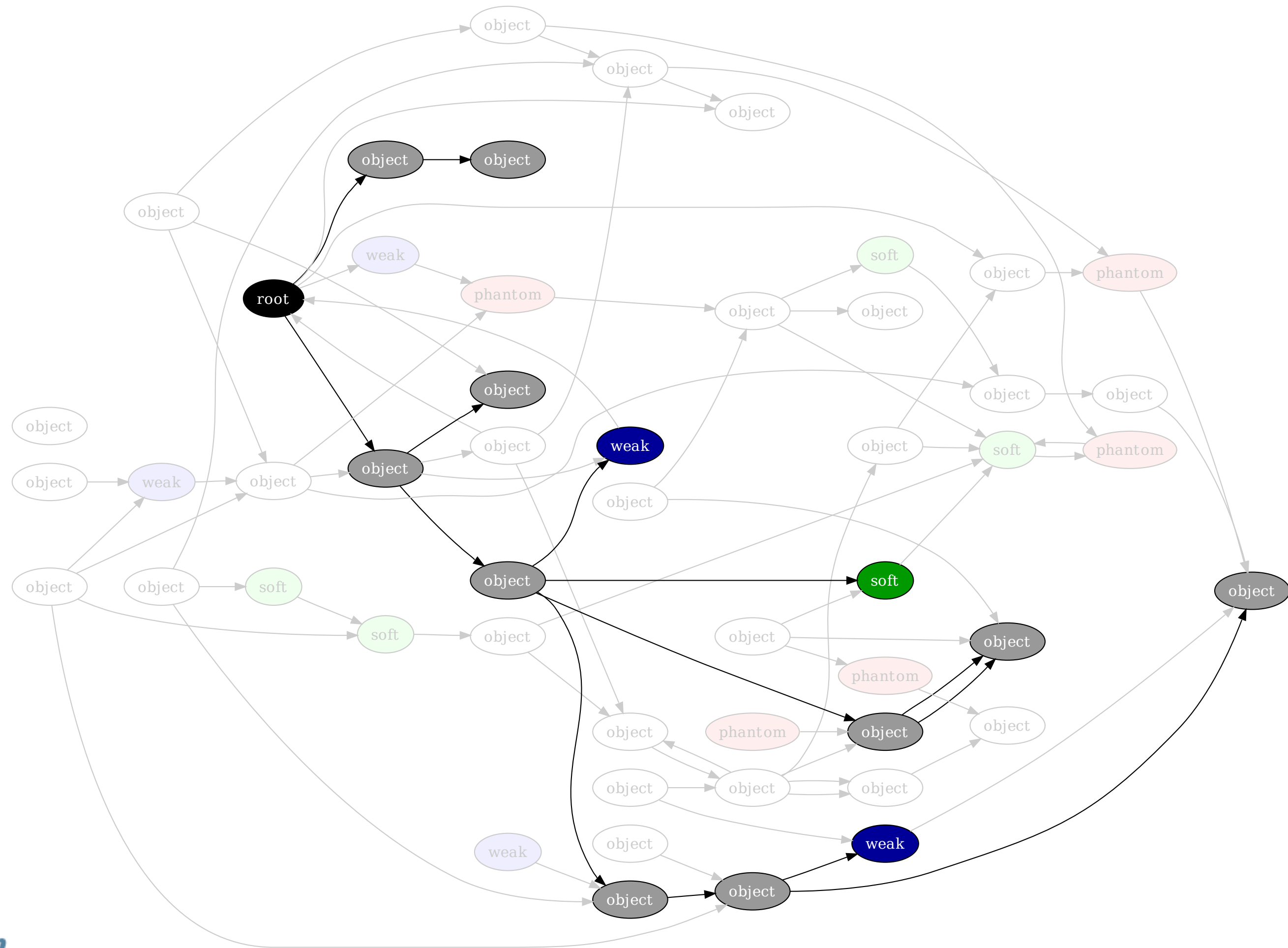
2. Start at a root.



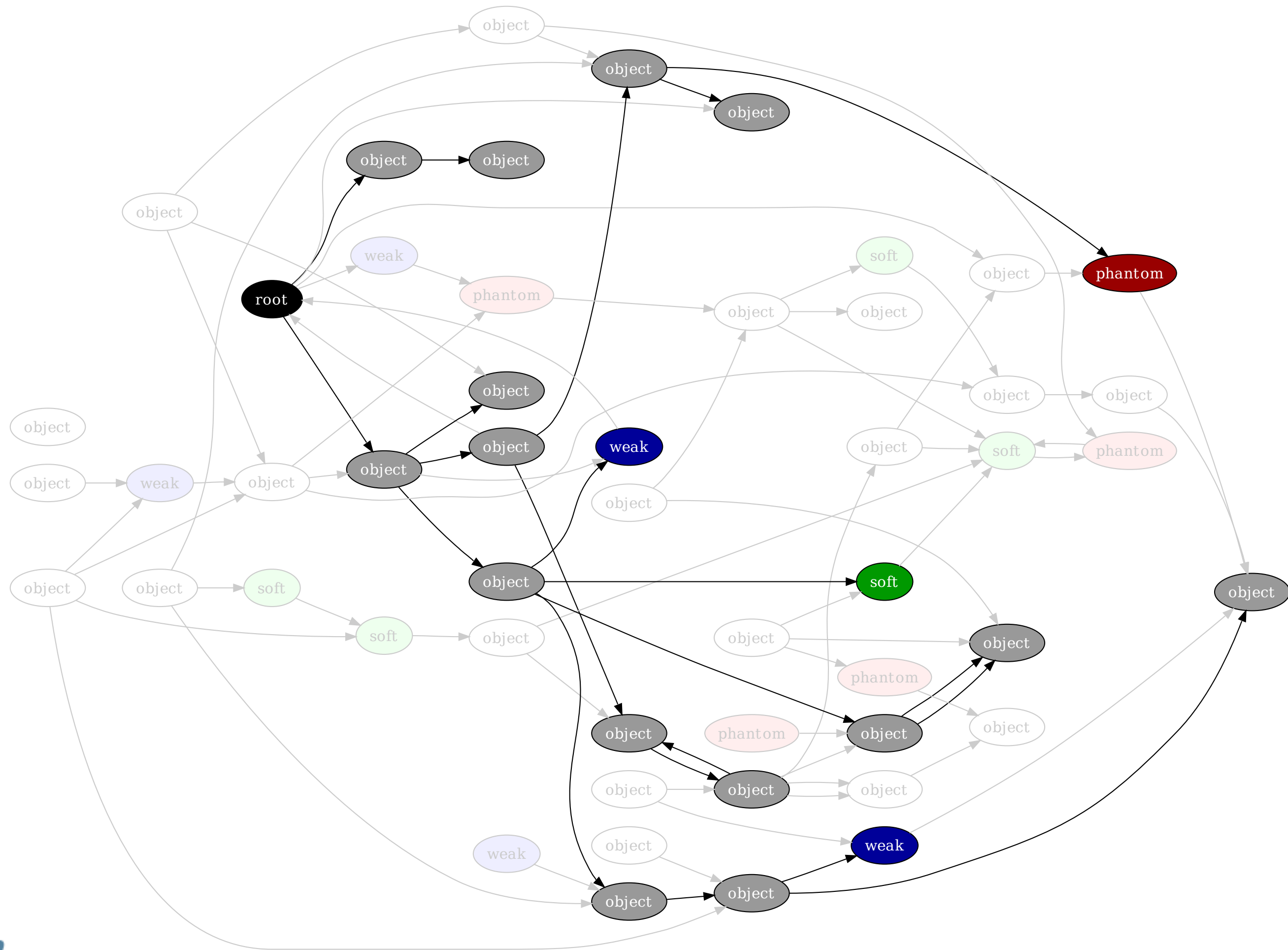
3. Trace & mark strongly-referenced objects.



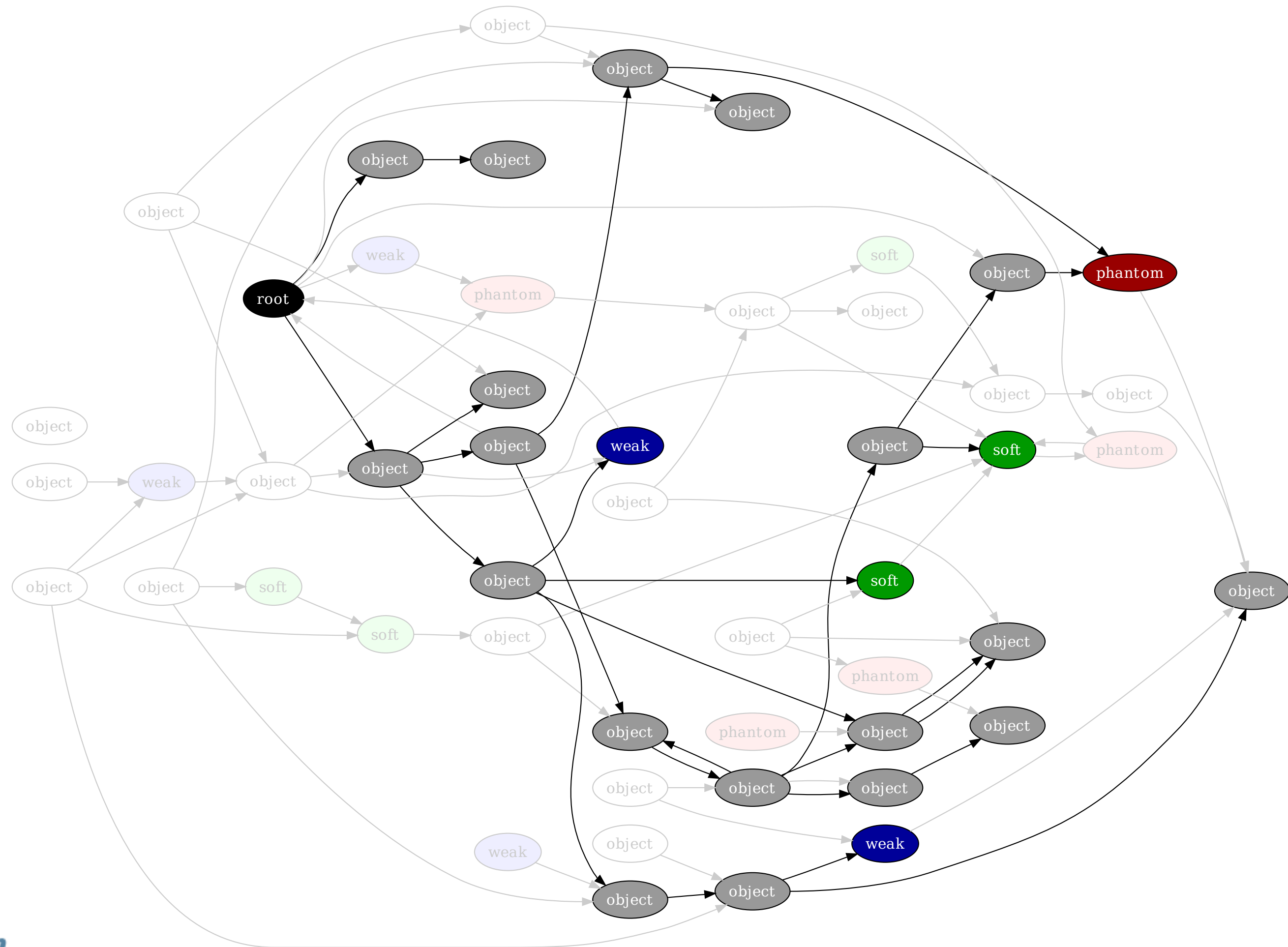
3. Trace & mark strongly-referenced objects.



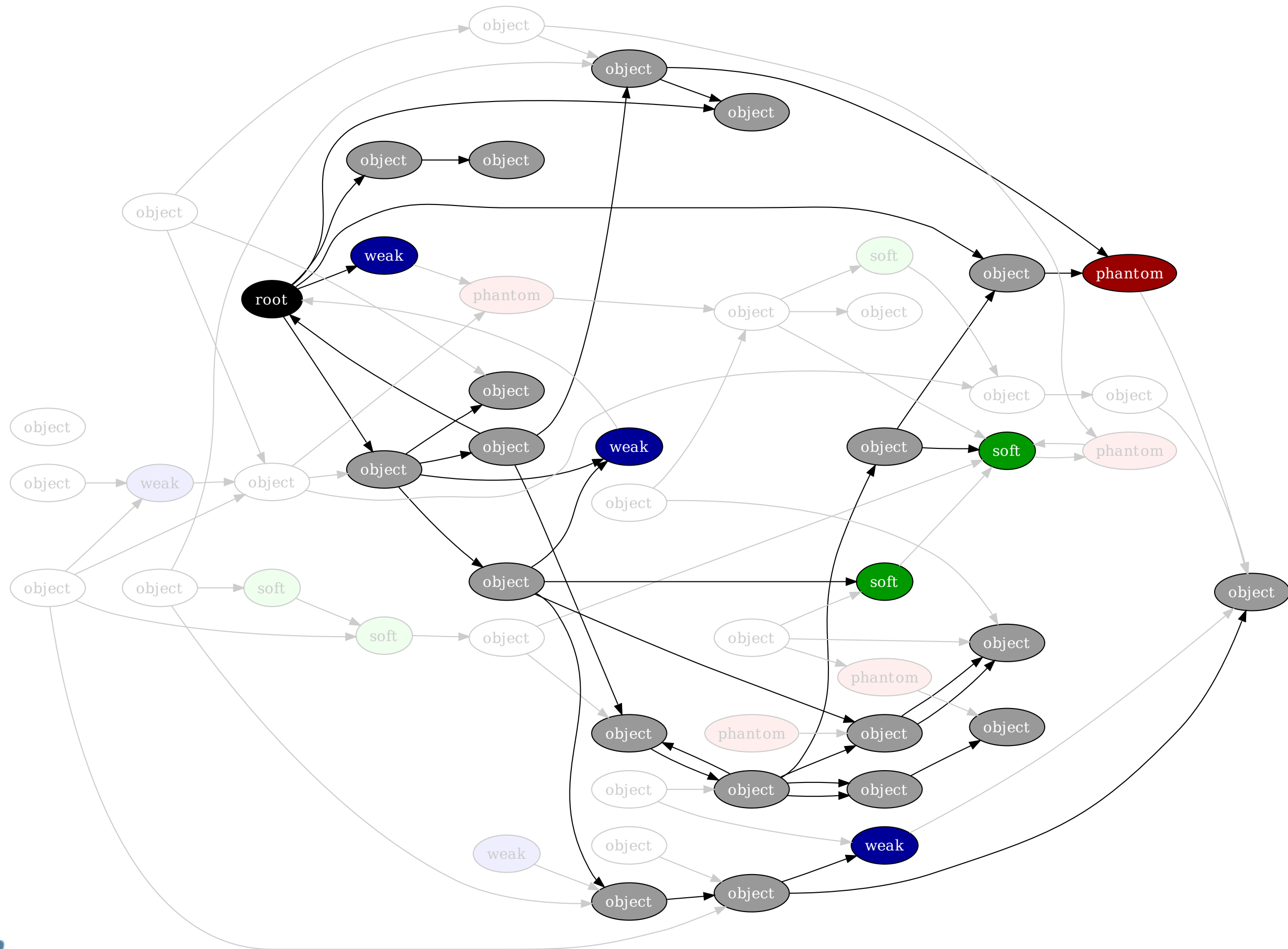
3. Trace & mark strongly-referenced objects.



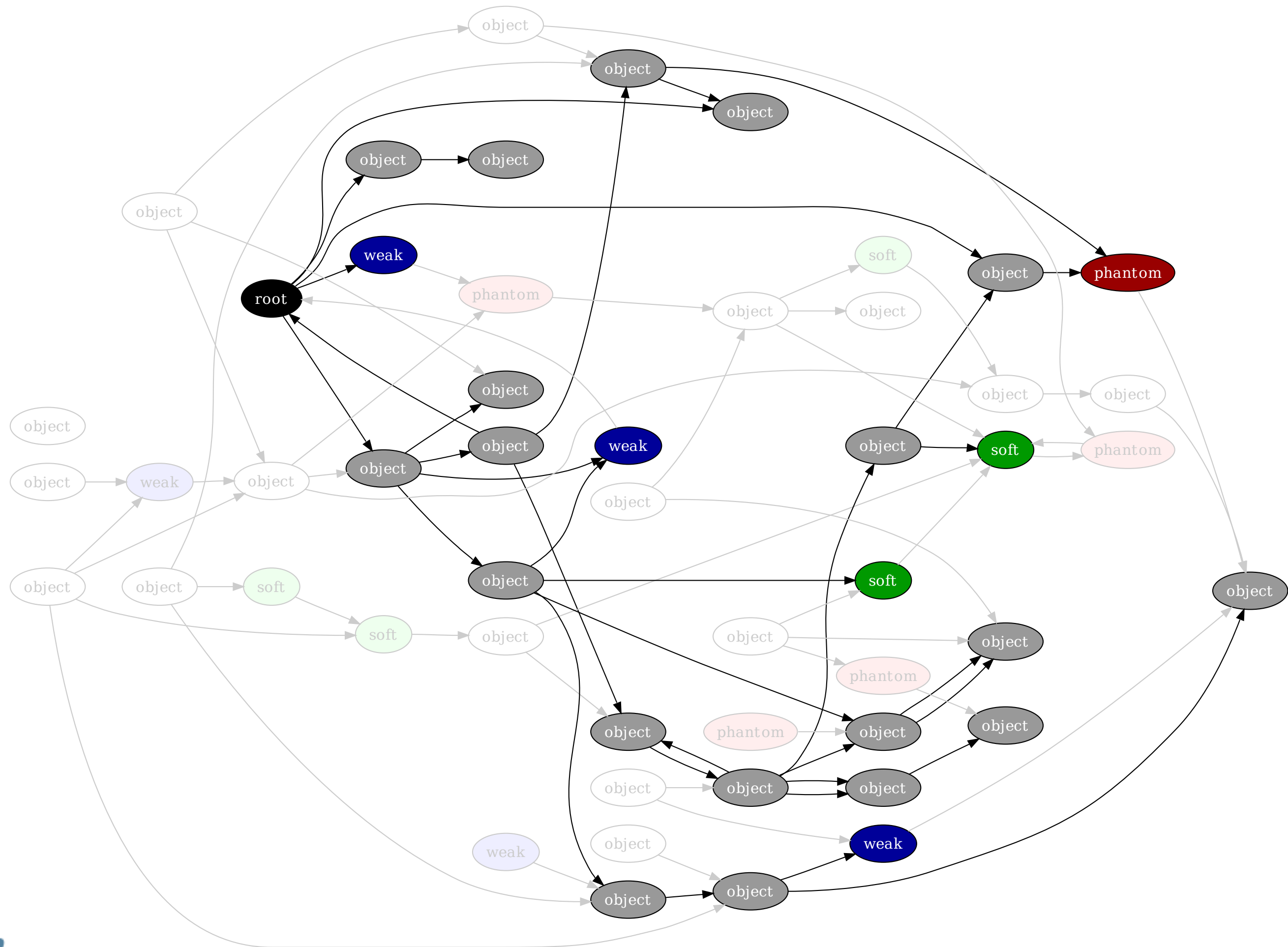
3. Trace & mark strongly-referenced objects.



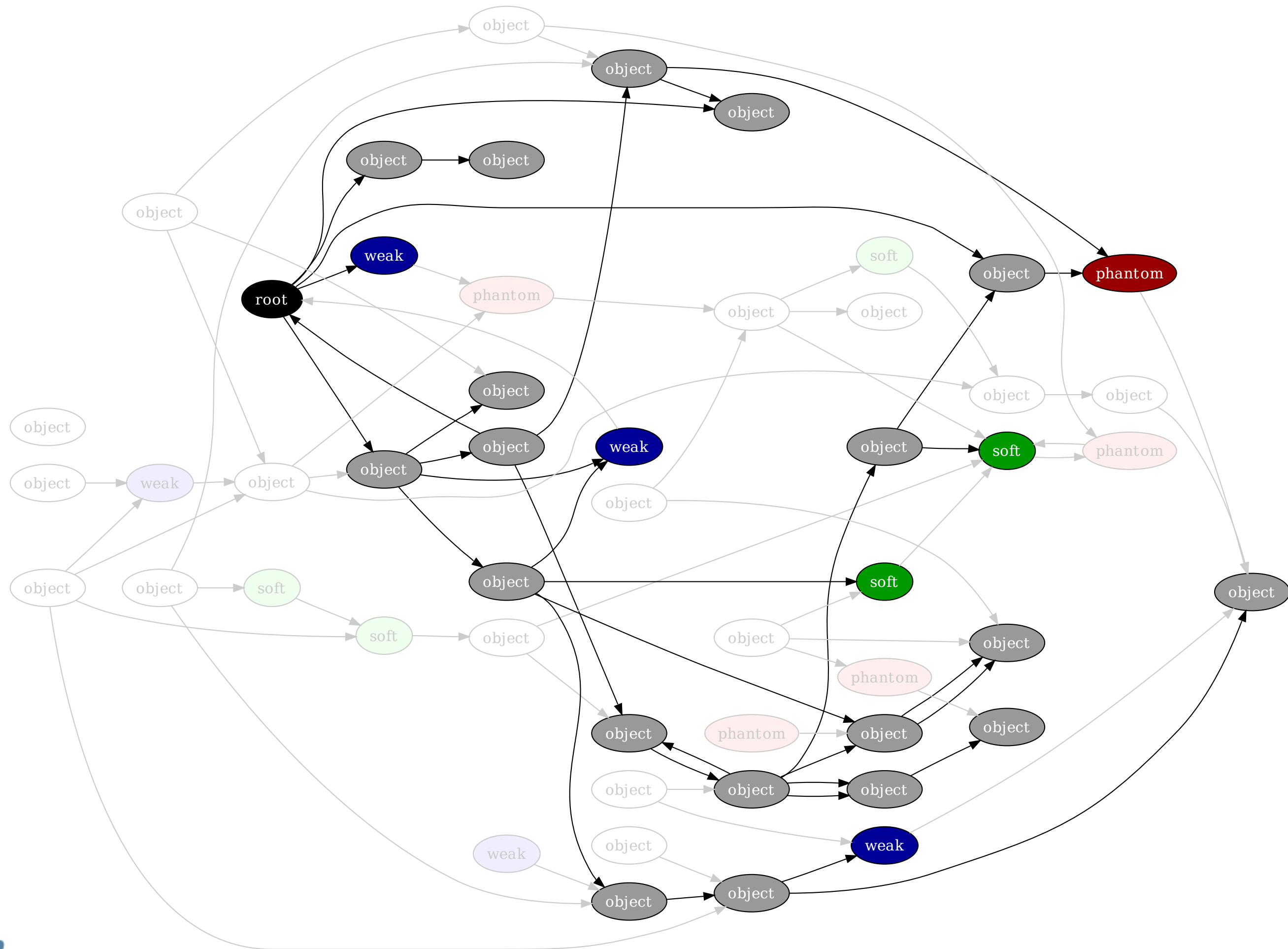
3. Trace & mark strongly-referenced objects.



4. Optionally clear soft references.



3. Trace & mark softly-referenced objects.



3. Trace & mark softly-referenced objects.

