Java is a trademark of Sun Microsystems, Inc.

Java

JavaOne℠

The Ghost in the Virtual Machine
A Reference to References

Bob Lee

Google Inc.

Google

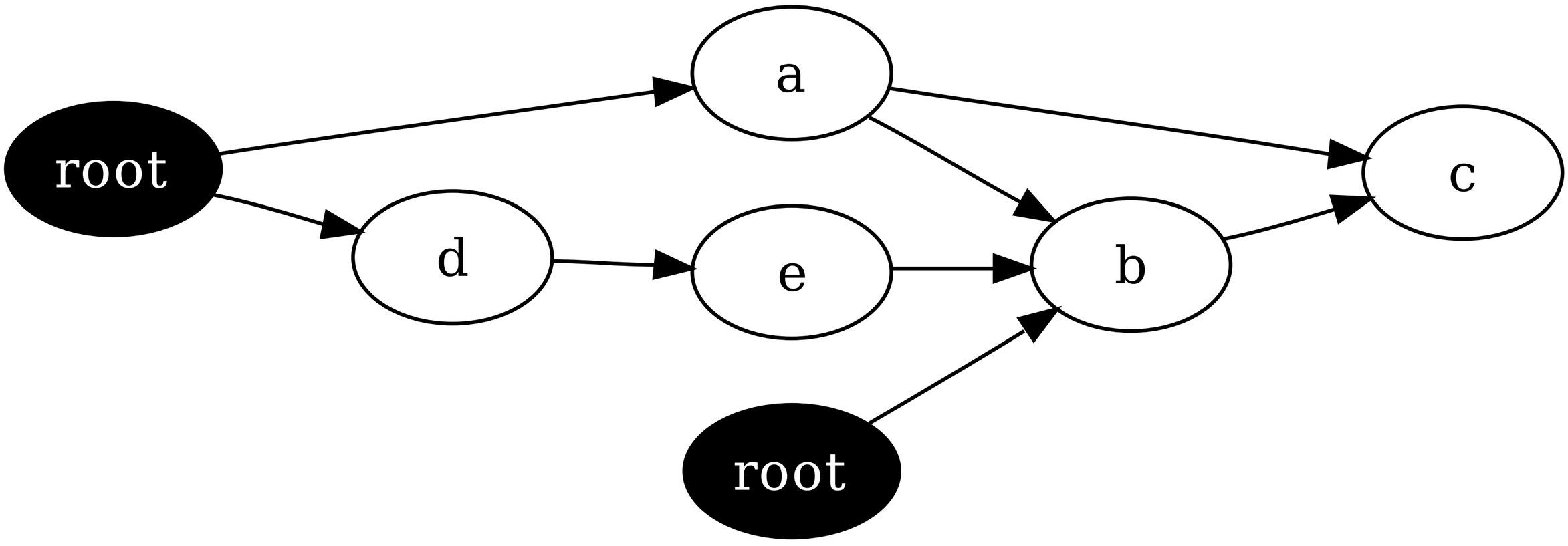
Goals

- Perform manual cleanup the right way.
- Take the mystery out of garbage collection.
- Become honorary VM sanitation engineers.

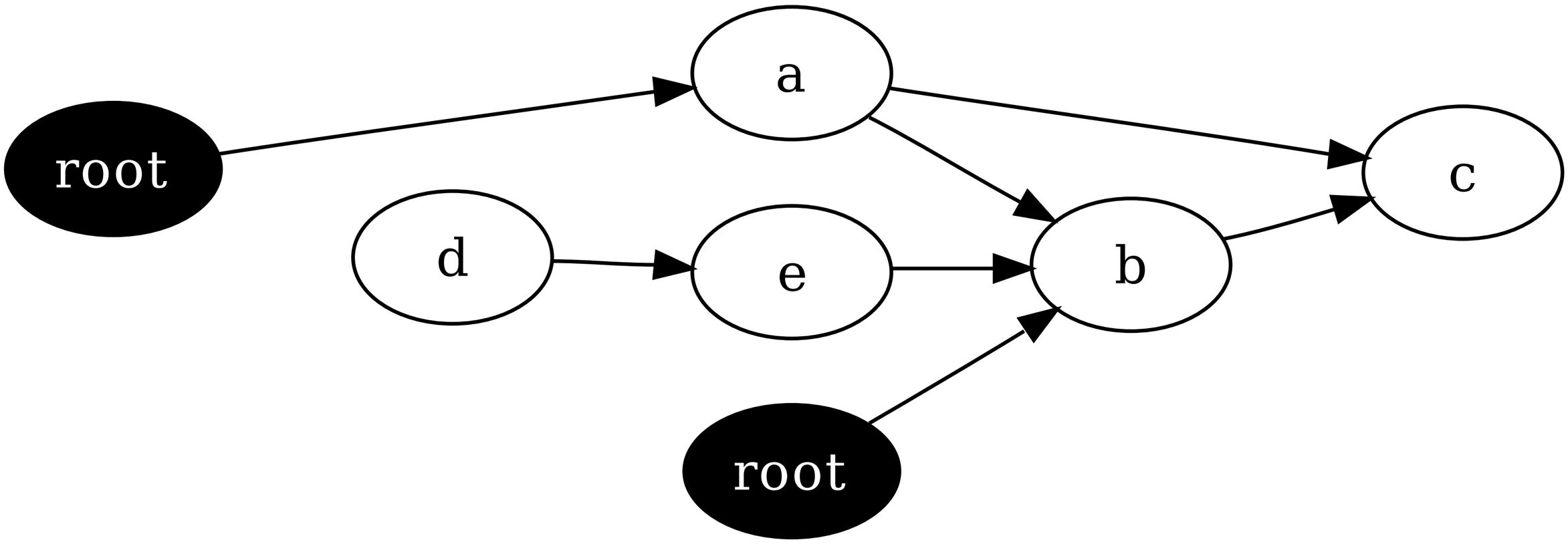
Code Example

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

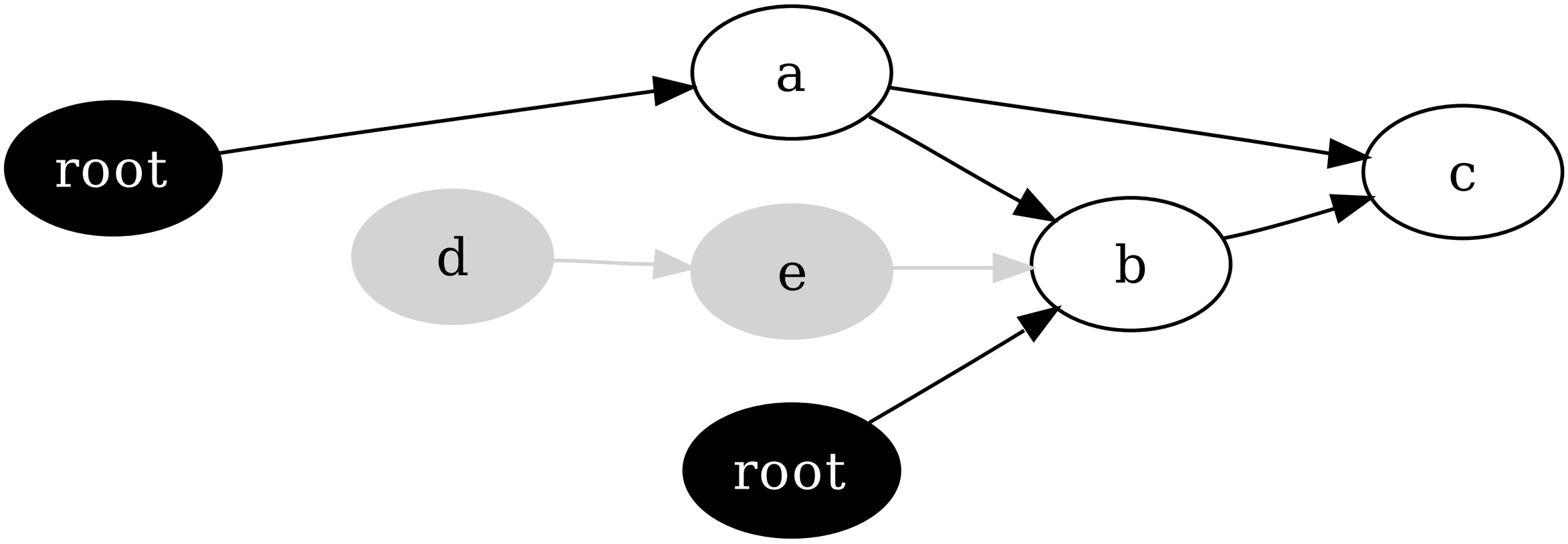
How does garbage collection work?



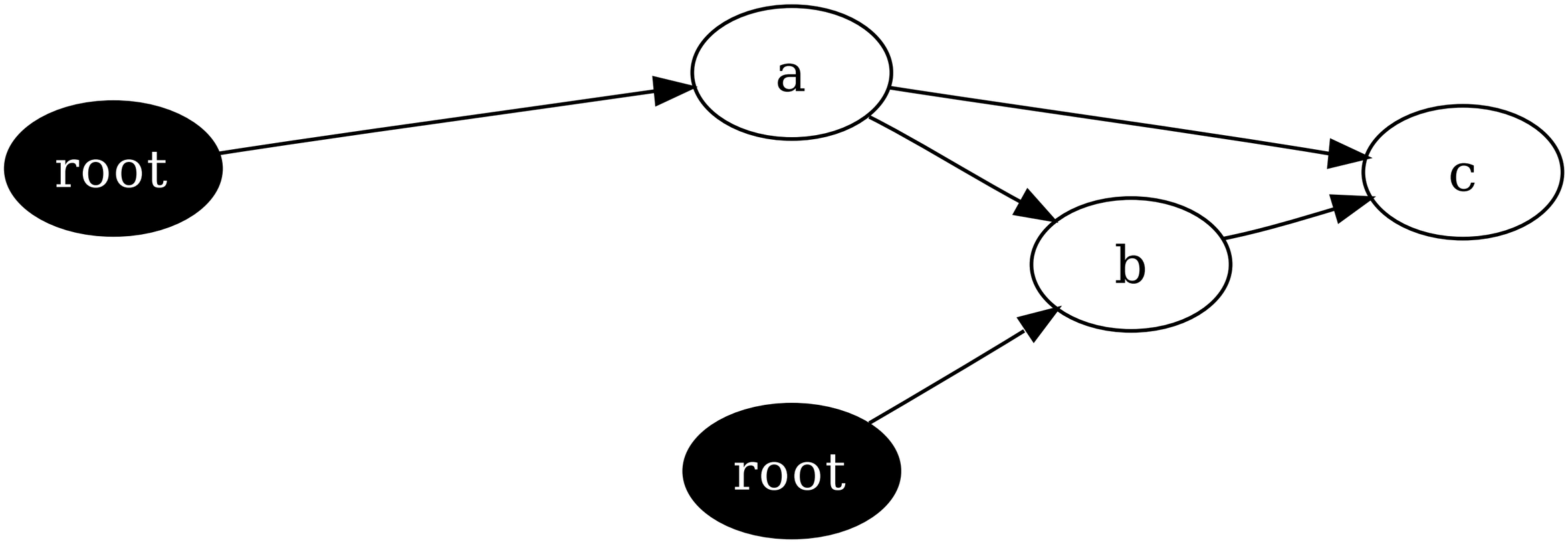
If the reference to D goes away...



We can no longer reach D or E.



So the collector reclaims them.



Reachability

- An object is *reachable* if a live thread can access it.
- Examples of heap roots:
  - System classes (which have static fields)
  - Thread stacks
  - In-flight exceptions
  - JNI global references
  - The finalizer queue
  - The interned String pool
  - etc. (VM-dependent)

Dante's Heap - The Levels of Reachability

- Strong
- Soft
- Weak
- Finalizer
- Phantom, JNI weak
- Unreachable

Dante's Heap - The Levels of Reachability

- **Strong**
- Soft
- Weak
- Finalizer
- Phantom, JNI weak
- Unreachable

Dante's Heap - The Levels of Reachability

- Strong
- **Soft**
- Weak
- Finalizer
- Phantom, JNI weak
- Unreachable

Dante's Heap - The Levels of Reachability

- Strong
- Soft
- **Weak**
- Finalizer
- Phantom, JNI weak
- Unreachable

Dante's Heap - The Levels of Reachability

- Strong
- Soft
- Weak
- **Finalizer**
- Phantom, JNI weak
- Unreachable

Dante's Heap - The Levels of Reachability

- Strong
- Soft
- Weak
- Finalizer
- **Phantom, JNI weak**
- Unreachable

Dante's Heap - The Levels of Reachability

- Strong
- Soft
- Weak
- Finalizer
- Phantom, JNI weak
- **Unreachable**

Two options for freeing native resources

> Use a finalizer.
  - You must defend against subsequent use!
> Or use a phantom reference.

Weak references aren't for caching!

- Many collectors will reclaim weak refs immediately.
- Use soft reference for caching, as intended:

"Virtual machine implementations are encouraged to bias against clearing recently-created or recently-used soft references."

- The `SoftReference` Javadocs