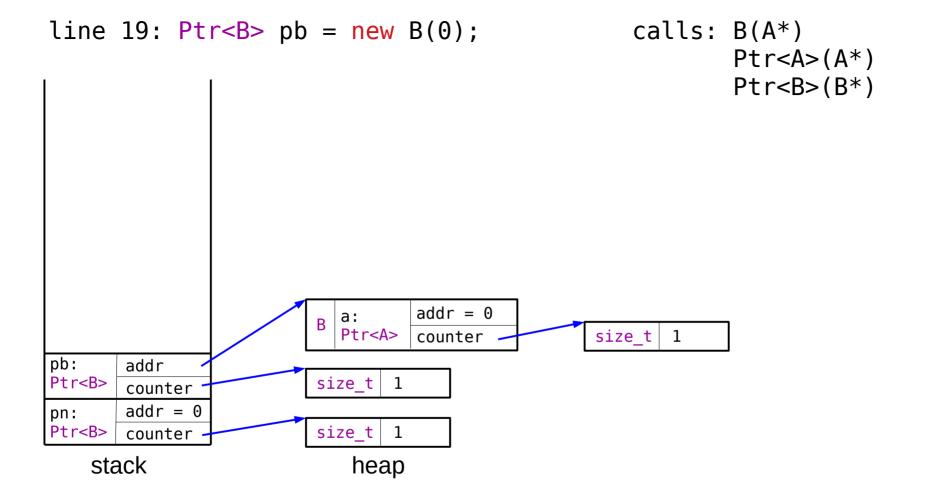
line 16: stack heap

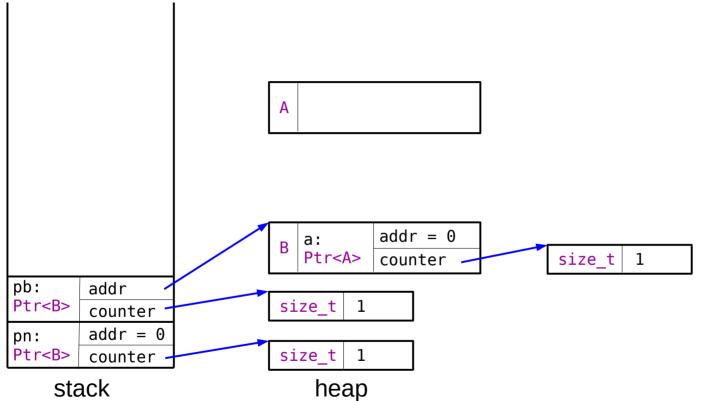
```
line 17: Ptr < B > pn = 0;
                                                  calls: Ptr<B>(B*)
      addr = 0
pn:
                       size_t 1
Ptr<B>
      counter -
   stack
                          heap
```

```
line 19: Ptr < B > pb = new B(0);
                                                  calls: B(A*)
      addr = 0
pn:
                       size_t 1
Ptr<B>
      counter -
   stack
                          heap
```

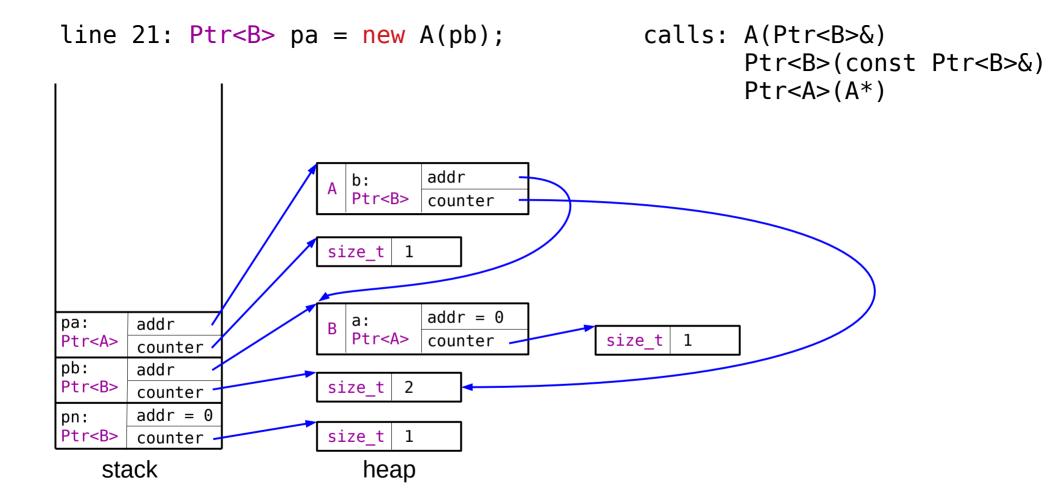
```
calls: B(A*)
line 19: Ptr < B > pb = new B(0);
                                                              Ptr<A>(A*)
                                 addr = 0
                          a:
Ptr<A>
                                                  size_t 1
                                 counter
       addr = 0
pn:
Ptr<B>
                        size_t | 1
       counter -
   stack
                           heap
```



line 21: Ptr pa = new A(pb); calls: A(Ptr&)



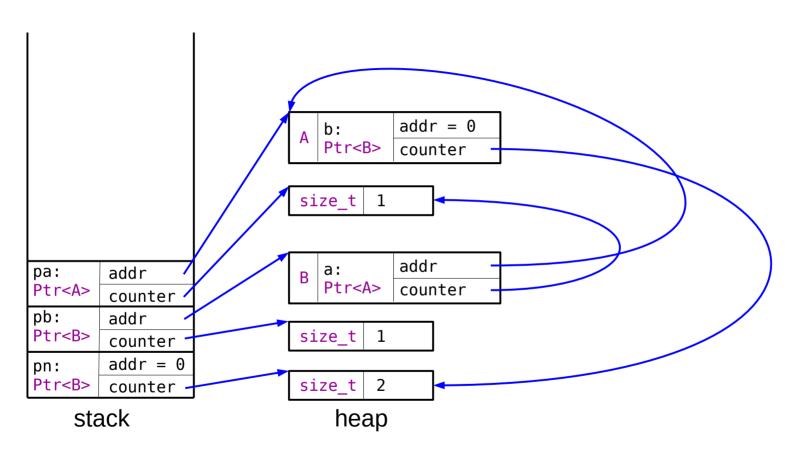
```
calls: A(Ptr<B>&)
line 21: Ptr < B > pa = new A(pb);
                                                              Ptr<B>(const Ptr<B>&)
                                 addr
                          b:
                          Ptr<B>
                                 counter
                                 addr = 0
                          a:
                          Ptr<A>
                                                  size_t 1
                                 counter
pb:
       addr
                        size_t 2
Ptr<B>
       counter
       addr = 0
pn:
                        size_t
Ptr<B>
       counter ·
   stack
                           heap
```



```
line 23: pb->a = pa;
                                                     calls: Ptr<B>::operator->()
                                                              Ptr<A>::operator=(..)
                                 addr
                          b:
                          Ptr<B>
                                 counter
                        size_t 2
                                 addr
                          a:
       addr
pa:
                          Ptr<A>
Ptr<A>
                                 counter
       counter
pb:
       addr
                        size_t
Ptr<B>
       counter
       addr = 0
pn:
                        size_t
Ptr<B>
       counter
   stack
                           heap
```

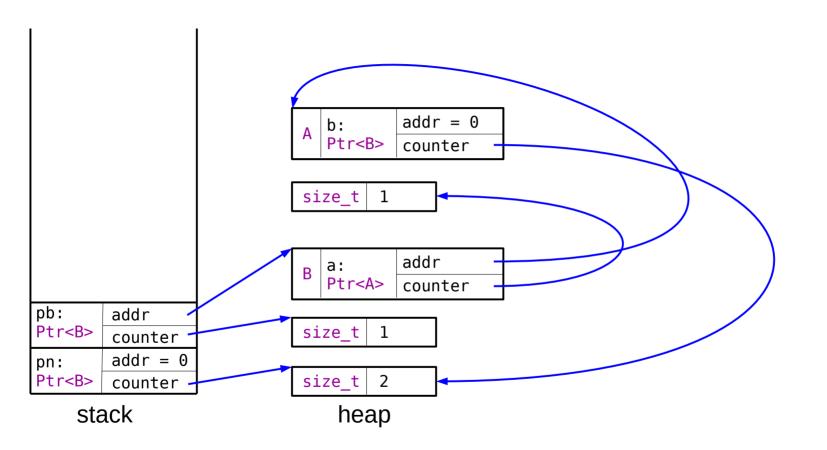
```
line 25: pa->b = pn;
                                                     calls: Ptr<A>::operator->()
                                                               Ptr<B>::operator=(..)
                                 addr = 0
                          b:
                          Ptr<B>
                                 counter
                        size_t
                                 addr
       addr
                          a:
pa:
                          Ptr<A>
Ptr<A>
                                 counter
       counter
pb:
       addr
                        size_t 1
Ptr<B>
       counter
       addr = 0
pn:
                        size_t
Ptr<B>
       counter
   stack
                           heap
```

line 27: return 0;

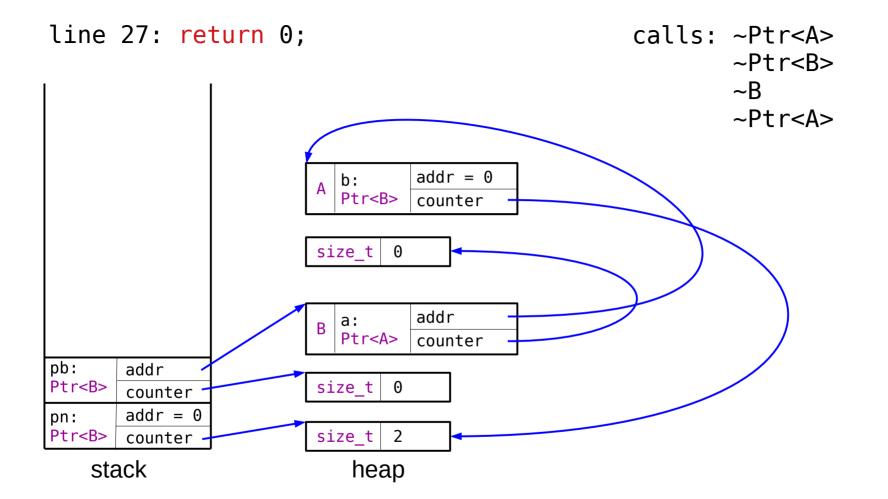


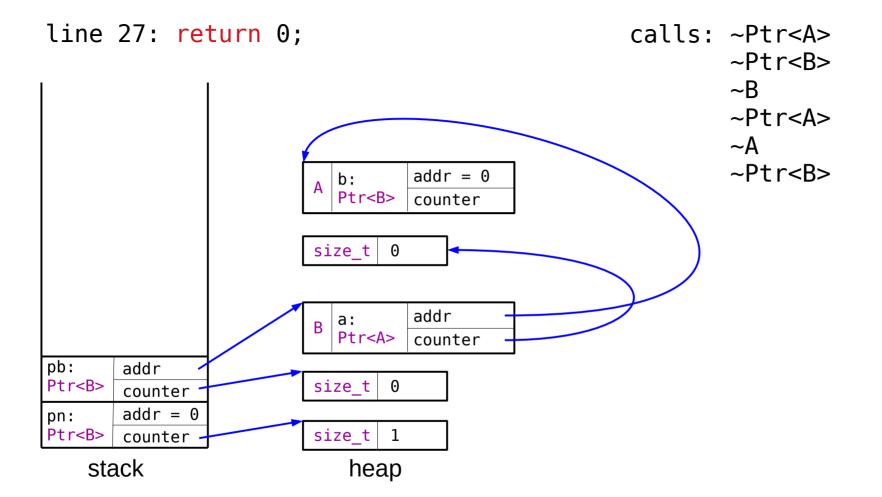
calls: ~Ptr<A>

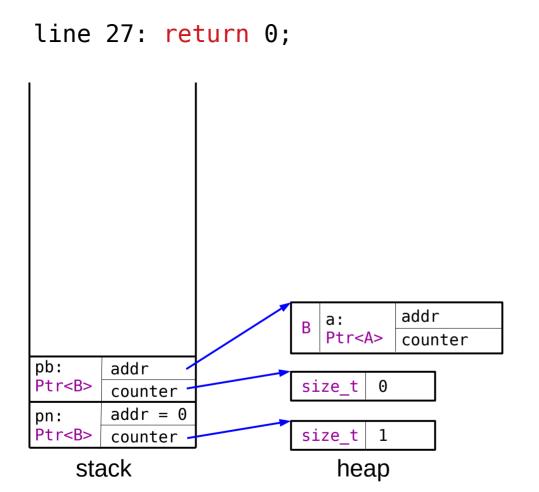
line 27: return 0; calls: ~Ptr<A>

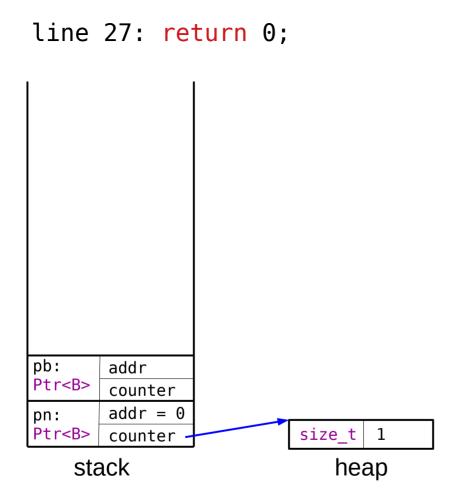


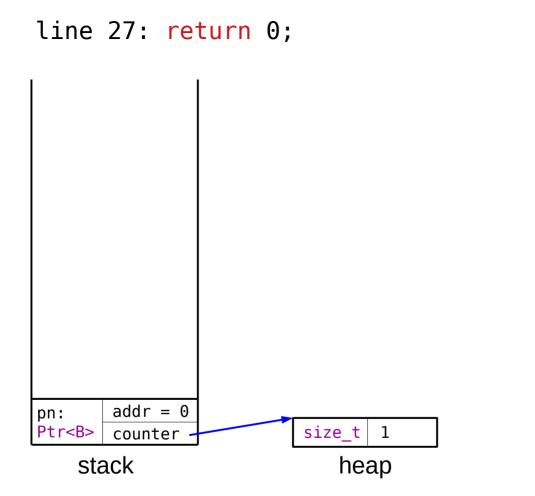
line 27: return 0; calls: ~Ptr<A> ~Ptr addr = 0b: Ptr counter size_t 1 addr a: Ptr<A> counter pb: addr size_t Ptr counter addr = 0pn: size_t Ptr counter · stack heap











line 27: return 0; calls: ~Ptr addr = 0pn: Ptr size_t 0 counter stack heap

```
line 27: return 0;
                                                calls: ~Ptr<B>
      addr = 0
pn:
Ptr<B>
      counter
   stack
                         heap
```

line 28: stack heap