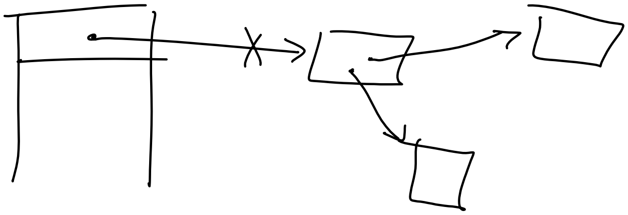
Memory Management

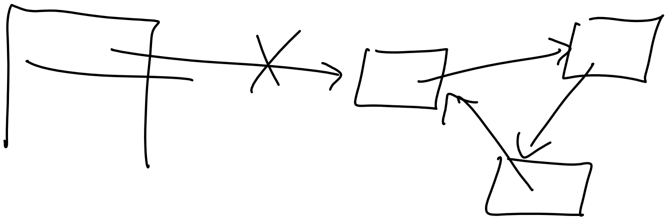
Using C w/ malloc & free.

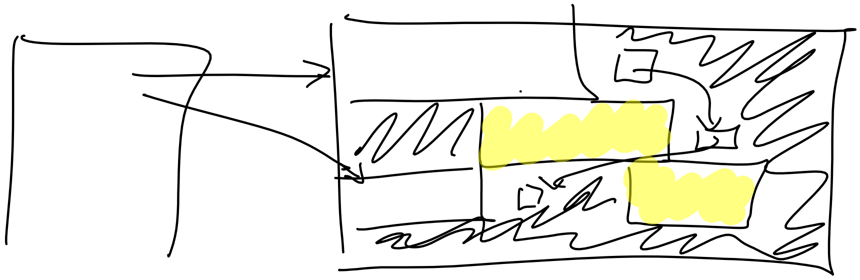
Reference Counting if last copy of a ptr is lost, free the memory it points to

shared_ptr<Foo> fptr =make_shared <Foo>(...);



```
class shared_ptr<T> {
   private:
      T* obj;
      int* count;
   ...
}
```





Assume malloc: FBR

1. Mark
2. Free anything not marked

Mark & sweep

Mark & don't sweep

1. Mark everything with a code (e.g. 1 bit)
2. When you need memory, allocate a block w/ no code (0 bit)
3. On next GC, just switch code

Generational GC

Generational hypothesis: most allocations are short-lived