

# Skippify (Spring '17, MT1)

We have the following `IntList` class, as defined in lecture and lab, with an added `skippify` function. Suppose that we define two `IntLists` as follows.

```
1 IntList A = IntList.list(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
2 IntList B = IntList.list(9, 8, 7, 6, 5, 4, 3, 2, 1);
```

Fill in the method `skippify` such that the result of calling `skippify` on A and B are as below:

- After calling `A.skippify()`, A: (1, 3, 6, 10)
- After calling `B.skippify()`, B: (9, 7, 4)

```
1 public class IntList {
2     public int first;
3     public IntList rest;
4
5     @Override
6     public boolean equals(Object o) { ... }
7     public static IntList list(int... args) { ... }
8
9     public void skippify() {
10        IntList p = this;
11        int n = 1;
12        while (p != null) {
13
14            IntList next = _____;
15
16            for (_____ ) {
17
18                if (_____ ) {
19
20                    _____
21                }
22
23                _____
24            }
25
26            _____
27
28            _____
29
30            _____
31        }
32    }
33 }
```