

Analyzing a non-trivial algorithm

```
void send(int* to, int* from, int count)
// Duff's device.  Helpful comment deliberately deleted.
{
    n = (count+7)/8;
    switch (count%8) {
    case 0: do { *to++ = *from++;
    case 7:  *to++ = *from++;
    case 6:  *to++ = *from++;
    case 5:  *to++ = *from++;
    case 4:  *to++ = *from++;
    case 3:  *to++ = *from++;
    case 2:  *to++ = *from++;
    case 1:  *to++ = *from++;
    } while (n>0);
    }
}
```

- The argument `count` implies that the source (`from`) and the destination (`to`) are arrays of integers, and `count` indicates the number of integers that needs to be copied (`send`) to the destination (`to`) array.
- `(count + 7)` resembles to a positive offset with *one byte* thus dividing it by 8 (one byte) results in the number of bytes, so `n` is the number of bytes.
- `switch (count%8)` if `count` was 10 then it would result to 2. This way `n` would also result to 2
($\Rightarrow (16 + 7)/8 = 2$)
($\Rightarrow 16 \bmod 8 = 0$)
- The do-while loop operates *only if count is the multiply of 8* and while `n` is not equal to 0.
Inside the loop as soon as a corresponding case is found (0 to start the loop), all the remaining cases are being executed beginning with (`case 7: *to++ = *from++;` to (`case 1: *to++ = *from++;`).

Conclusion

The algorithm above *sends messages* represented as integers in `count` packets from destination `from` to destination `to`. `Count` corresponds to the number of packets, whereas one packet must be the multiply of 8.

A packet is being sent in every iteration.