

















## Merge sort

- But, how do we sort the two haves of the list?
  - By splitting them in half, sorting the halves and merging of course
  - Eventually we reach a base case with 1 element

```
void msort(int data[]) {
    msort(data,new int[data.length],0,data.length-1);
}
void msort(int data[],int extra[],int first,int second) {
    if (data.length>1) {
        middle=(first+second)/2;
        msort(data,extra,first,middle-1);
        msort(data,extra,first,middle,second);
        merge(data,extra,first,middle,second);
    }
}
```

```
void merge(int data[],int extra[],int first,int second,int last) {
    int i,b1,b2;
    b1 = first; b2 = second;
    for (i=0;i<(last-first+1);i++) {
        if (b1<second && (b2>last || data[b1]<=data[b2])) {
            extra[first+i]=data[b1];
            b1++;
        } else {
            extra[first+i]=data[b2];
            b2++;
        }
    }
    for (i=0;i<(last-first+1);I++) {
        data[first+i]=extra[first+i];
    }
}</pre>
```

