ForkJoin Framework

Method	Use
`compute`	Thread objects override (void/V) method `compute` When `fork` is called, `compute` method is executed A bit like main(…)---default starting point
`fork`	Starts a new thread executing `compute` method
`join`	Calling `otherThread.join()` pauses this thread until `otherThread` has completed its `compute`.
`RecursiveTask<V>`	Class which we extend to make threads to return a result of type `V`. `compute` returns `V` for this object
`Recursive Action`	Class we extend when we don't return a result
`ForkJoinPool`	Object that manages threads
`invoke`	Pool method to start first thread object.

14

```
…
protected Integer compute(){
      if(hi-lo < cutoff){
        int ans=0;
        for(int i=lo; i<hi; i++)
           ans += arr[i];
           return new Integer(ans);
      }
      else{
        SumThread left = new SumThread(arr, lo, (hi+lo)/2);
        SumThread right = new SumThread(arr, (hi+lo)/2, hi);
        left.fork();
        Integer rightAns = right.compute();
        Integer leftAns = left.join();
        return newInteger(leftAns + rightAns);
      }
}
…
```

19

Reduce

It shouldn't be too hard to imagine how to modify this code to:

1. Find the maximum element in an array.
2. Determine if there is an element meeting some property.
3. Find the left-most element satisfying some property.
4. Count the number of elements meeting some property.
5. Check if elements are in sorted order.
6. [And so on…]

22

Useful Diagram

One node per O(1) operation

What does the dependency graph look like for that snippet?

```
01: x=x+5
02: y=x+7
03: z=x+13
```

30