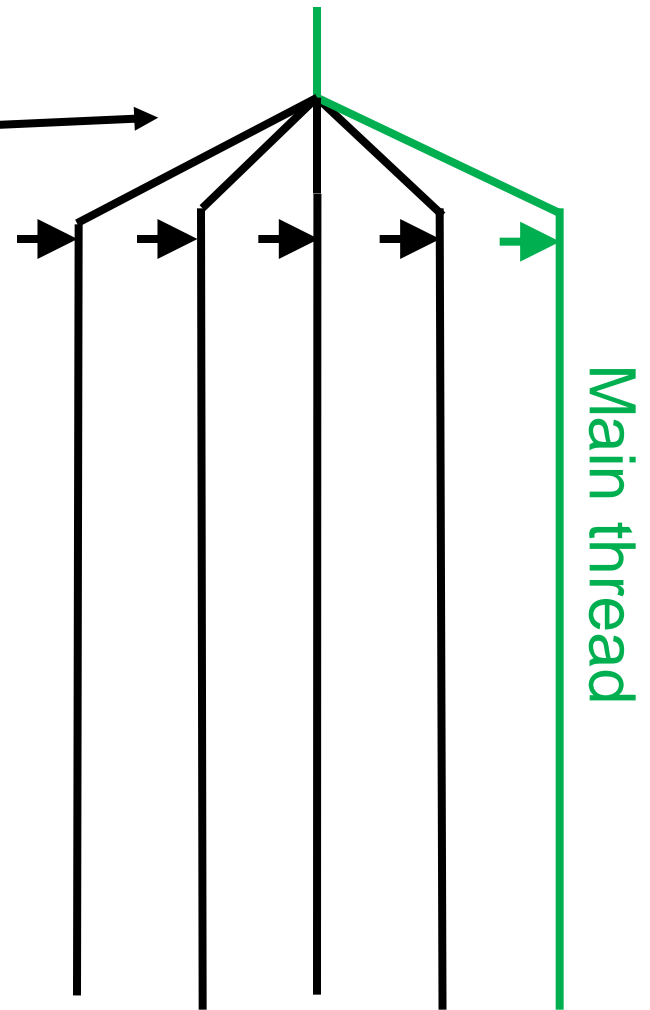


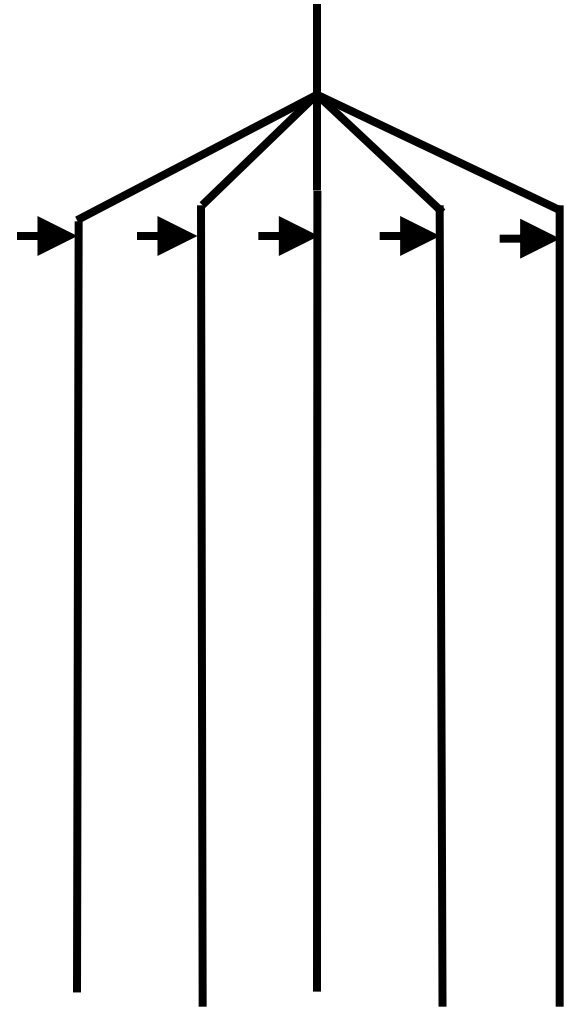
Executor Service

```
ExecutorService tpe =  
Executors.newFixedThreadPool(4);
```



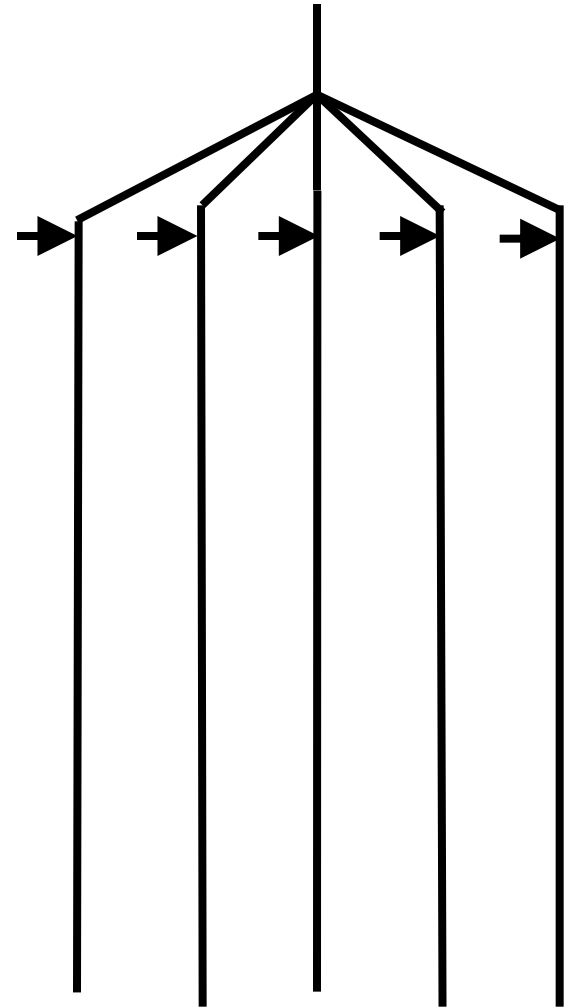
Executor Service

```
public class MyRunnable implements Runnable {  
    int a;  
    ExecutorService tpe;  
  
    public MyRunnable(ExecutorService tpe, int a) {  
        this.a = a;  
        this.tpe = tpe;  
    }  
  
    @Override  
    public void run() {  
        if (a > 10) {  
            tpe.shutdown();  
            return;  
        }  
        System.out.println(a);  
        tpe.submit(new MyRunnable(tpe, a + 3));  
    }  
}
```

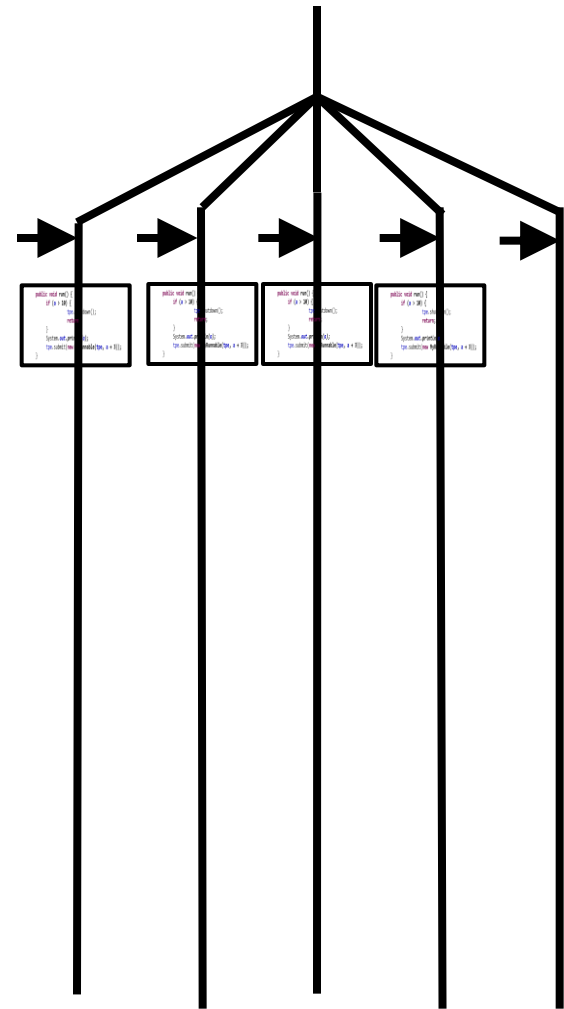


Executor Service

```
tpe.submit(new MyRunnable(tpe, 0));  
tpe.submit(new MyRunnable(tpe, 1));  
tpe.submit(new MyRunnable(tpe, 2));  
.....
```



Executor Service



```

public void run() {
    if (a > 0) {
        run();
    }
    System.out.println("a = " + a);
    a--;
}

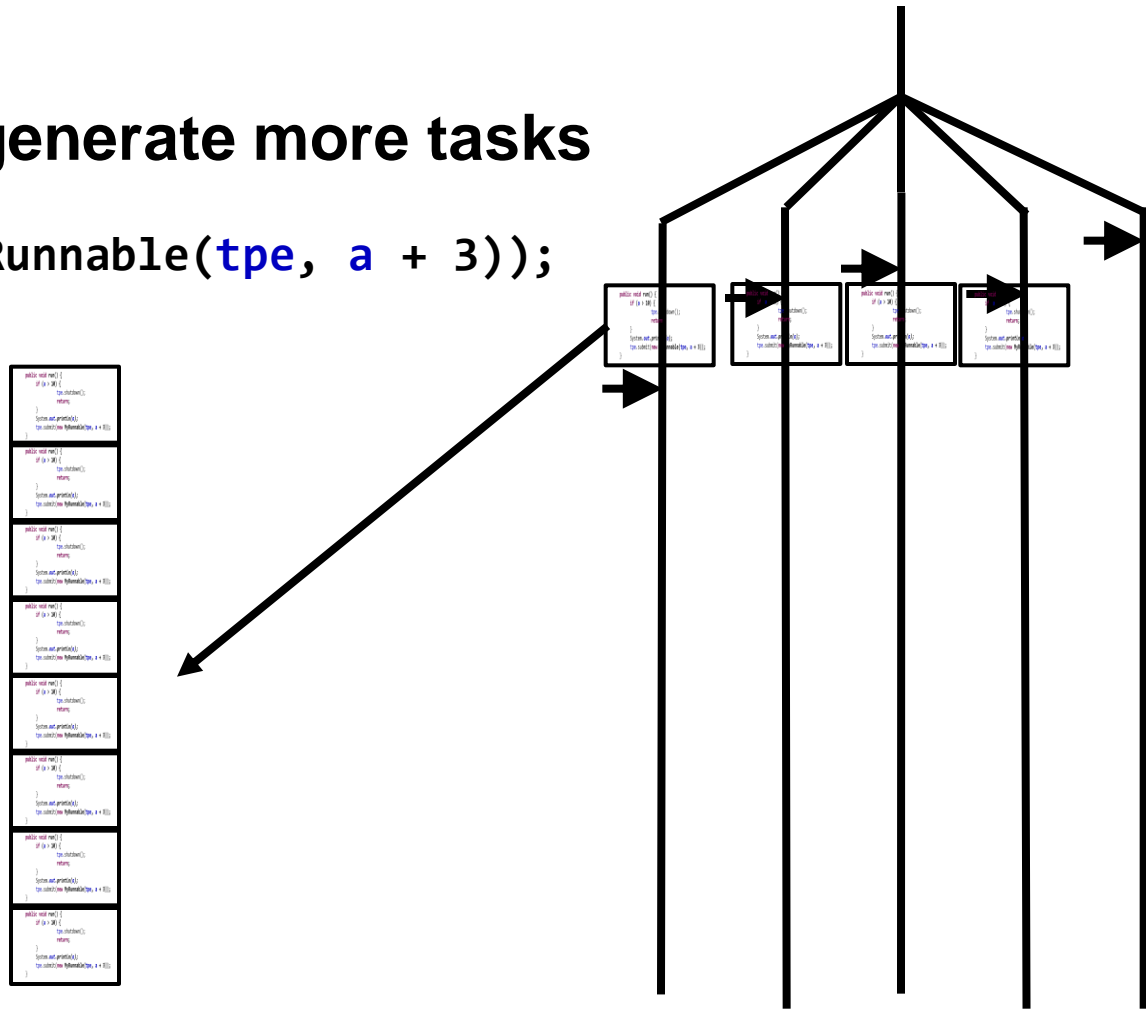
public void run() {
    if (a > 0) {
        run();
    }
    System.out.println("a = " + a);
    a--;
}

```

Executor Service

Tasks can generate more tasks

```
tpe.submit(new MyRunnable(tpe, a + 3));
```



Executor Service

```

public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

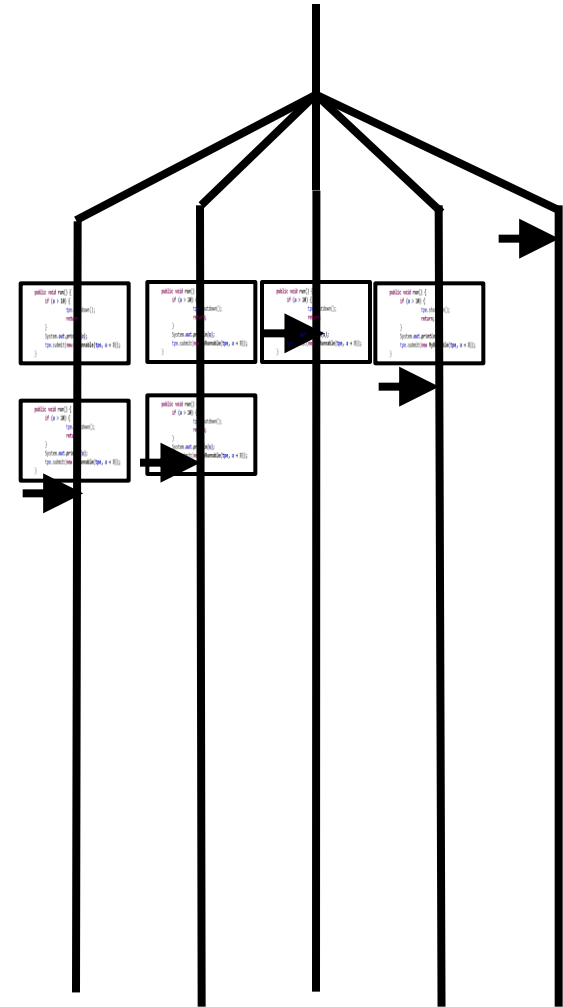
public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

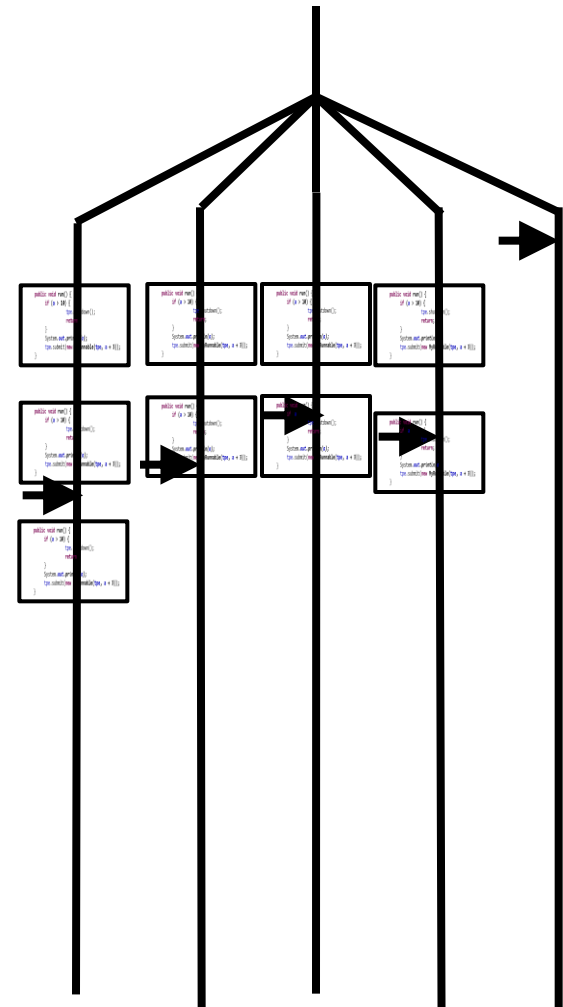
public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

public void run() {
    if (isRunning()) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
    System.out.println("Task " + taskName);
    taskName = "Task " + (taskName.length() + 1);
}

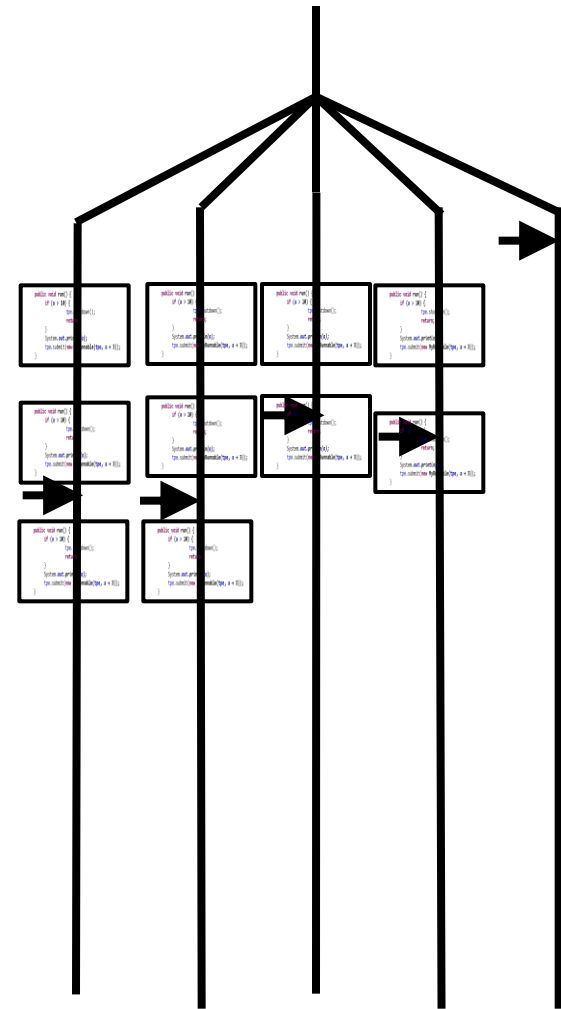
```



Executor Service



Executor Service



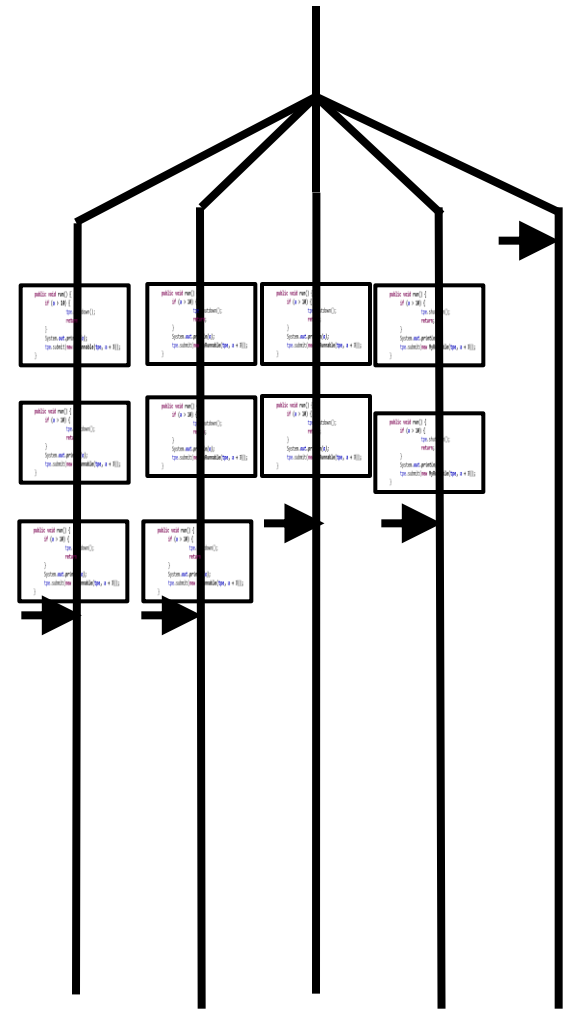
```
public void run() {
    if (i < 10) {
        try {
            sleep(1000);
        } catch (InterruptedException e) {}
        System.out.println(i);
        try {
            sleep(1000 * Math.random() * 10);
        } catch (InterruptedException e) {}
    }
}

public void run() {
    if (i < 10) {
        try {
            sleep(1000);
        } catch (InterruptedException e) {}
        System.out.println(i);
        try {
            sleep(1000 * Math.random() * 10);
        } catch (InterruptedException e) {}
    }
}

public void run() {
    if (i < 10) {
        try {
            sleep(1000);
        } catch (InterruptedException e) {}
        System.out.println(i);
        try {
            sleep(1000 * Math.random() * 10);
        } catch (InterruptedException e) {}
    }
}

public void run() {
    if (i < 10) {
        try {
            sleep(1000);
        } catch (InterruptedException e) {}
        System.out.println(i);
        try {
            sleep(1000 * Math.random() * 10);
        } catch (InterruptedException e) {}
    }
}
```


Executor Service



```

public void run() {
    if (isDone()) {
        return;
    }
    System.out.println();
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {}
}

public void run() {
    if (isDone()) {
        return;
    }
    System.out.println();
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {}
}

public void run() {
    if (isDone()) {
        return;
    }
    System.out.println();
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {}
}

public void run() {
    if (isDone()) {
        return;
    }
    System.out.println();
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {}
}
    
```

Executor Service

When do you stop the threads?
Depends on the problem.
Sometimes one solution is enough.
But some problems might not have solutions.

`tpe.shutdown();`

