Design

pipe

```
interface Stage:
* Handle(Item) Item: Handles the Item to be moved through the Pipe.
```

Stage corresponds to stages of computations in 1. Item refers to any type satsfying 6.

```
StageFunc func(Item) Item
```

StageFunc is a utility to convert functions which would satisfy Stage into Stages.

```
class Pipe([]Stage):
* Get(Item): Get puts the Item in the Pipe to be processes.
* Give() Item: Give blocks until an Item is done being processed and returns it.
```

Pipe connects Stages in the given order satisfying 4 and the rest of 1. Get and Give hide concurrency by immediately returning after putting an Item in the Pipe and blocking until an Item is done satisfying 5. Concurrent functions are started for every Stage whenever the Pipe has an Item that handles Items concurrently when available then places them in the next Stage satisfying 2 and 3. These functions exit when the Pipe is empty.

```
func Process(Pipe, []Item) []Item
```

Process is a utility to run many Items through a Pipe.