- [ ] Tier of batches
- [ ] batches of batch[]
- [ ] Batch of nodes
- [ ] node
- [ ] node names procedure:

~~(gonum package used).~~
integer entered "d"
characters are of n length '26'
thus min value is 1 and max value is 26 for first set, min value is 27 and max value is 26+26x26, making a general rule of total max is n^1 +n ^2 + ...+n^k
to calculate the node name of entered value "d":
if _min <= d<= _max where _min and _max correspond to min,max valies of the node variance.
loop for as long as (sub=(d -(min -1)) > min -1)
then the first letter symbol "s1" is the

number of loops of the array of characters indexed by 1.

when (sub < (min -1)) then recursively call the function again with sub as its input integer and append it to the string to be returned. try to make the recursive call a tail recursive thus minimizing the big0 complexity.

- saving and loading nodes should go within a yaml file of the following structure:

```
batch:
    sampling_date: time.now()
    name:"test1"
    info:"some info"
    comment:"comment"
    threshold: 3.0
    precision:2
nodes:
```

- node:
  name :"A"
  ir:18.00
  ar:18.00
  n:1