

# CLA Tips

Code advices:

- DO NOT alter function names, input types in `cla_utils`
  - you may change output and docstring if you feel that is required
- swap with the following code  
`a, b = b, a`
  - other ways are inefficient
- Try to use np array instead of list, as arrays access are more efficient (see file *Making Python Faster* for details)
  - but use of `range(n)` acceptable
  - do NOT build list inside list.
  - list comprehension is not recommended
- Minimise number of for loops, use matrix slicing instead of for loop (it seems that only one loop is allowed for most cases)
  - But be aware that matrix slicing is SHALLOW copy!!!
  - Also minimise other operations in for loop, e.g. if statement, unnecessary calculations that could be put outside the loop
  - see *Making Python Faster* for more advices
- NO printing, returning, producing figures in tests (PyTest functions)
- NO leaving old version of codes in comments, just delete them
- No need to check "if flag is not None", "if flag" is enough. (None are treated as False)
- put any non-function codes in exercises under  
`if __name__ == "__main__":`
  - to avoid them run every time the module is imported
- Carefully check useless files are not pushed to GitHub
  - Only push .py files, data, figures etc.
  - e.g. `__pycache__`, `.pytest_cache`, `clavenv`, `egg_info`
- Docstring of own functions should be comprehensive:
  - Clearly explain meaning of every input variable, what the function will do on input and meaning of output variable
  - using the format `:param x: data type of x`, clearly write data types of all input variables
  - using the format `:return y: data type of y`, clearly write data types of all output variables
- `solve_U`, `solve_L` takes 2d array and returns 2d array, be careful !!
- To optimise storage, permutation matrices should NOT be built explicitly
- norm is inner product of `x` and `np.conj(x)`, it is recommended that you write this in a norm function and use it instead of `np.linalg.norm`. Do

check your norm function works

```
def norm(x):
```

```
    """
```

- update: use of `np.linalg.norm()` is now an approved exception of `linalg` package (see edstem string #185)

other advices:

- Please double check figures are corrected inserted to the report, and matches the caption.
  - All plots should have correct x, y axis labels, correct title. And if there are multiple lines/groups of points, use legend to distinguish them
- When comparing two methods, try to use both quantitative and qualitative arguments.