

cp-library

Rules

Allowed defines

- `ll -> long long`

Dependency handling

Generating output file with dependencies

In the generator directory is a node js script (`main.js`). The script has dependencies, which must be installed using npm or yarn. Run the following commands.

```
cd generator
npm install
```

Run the following command to execute the script (with path as your `*path*` to the `main.js` file.

```
node *path*/main.js
```

First input the category of the datastructure / algorithm which you want generated, then the actual datastructure / algorithm. The script will autocomplete your input if you press tab. After running, the generated output will be in your clipboard and in a generated file named `./temp-generated.cpp`. The script will recursively join files with dependencies.

Including dependencies

In order to denote that a given algorithm / datastructure depends on another you should, at the position in the `main.cpp` file where you want the dependency inserted, write the following.

```
%include<category/concept>%
```

`category` is a folder name in `src`, and `concept` is a folder name in `src/category`. Here is an example.

```
// your code
```

```
%include<data-structures/union-find>%
```

```
// your code
```