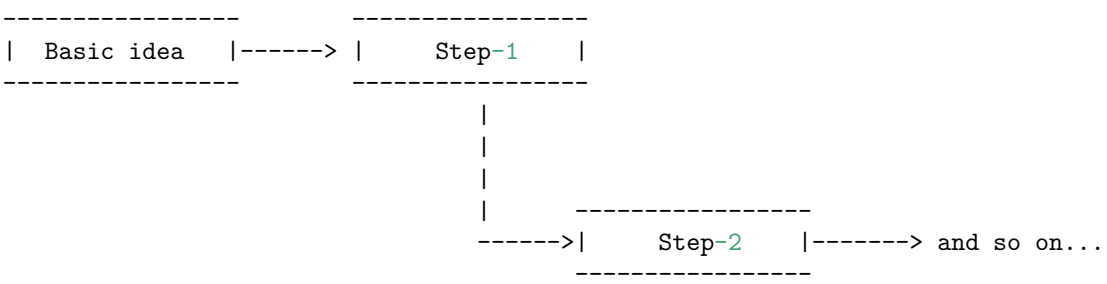
Flow (idea flow tracker)

- Need to think of a simple name, for time being lets call it `flow`.
- A simple terminal ncurses based application to track research ideas and the possible options that we take in the way to achieve conclusion.
- The flowchart blocks should be able to link to the code location.
- This is not supposed to be a simple terminal flow chart application.
- Upon clicking or pressing enter the block should open the full description otherwise it should just display the summary.
- The plan is store the data in json/sql/plain text format whichcould be parsed by other GUI/web clients.
- This tool could be part of any code repo.
  - This tool can be initiated like git init.
- The code needs to be written with modularity and reusability in mind.
  - Maybe this needs to be broken down into simpler projects.
  - Like the idea drawer which will parse the json files and draw a graph could be a project in itself.
  - **This needs more thought.**

Flow chart representation



Need to create a better flow diagram in xfig

- **Need to draw a state machine for the idea editor flow**.
  - Draw a state machine catering to all decided shortcut keys.

Idea struct

```
struct idea_node{
    int idea_id; /* Store the global idea id in the head and copy it in the branchouts */
    char summary[MAX_SUMMARY_LENGTH]; /* Store the summary, it should be 80 chars long */
    char description[MAX_DESCRIPTION_LENGTH]; /* Store the description of the step */
    struct idea_node *branchouts[MAX_BRANCHOUTS]; /* Store the pointer to the branchout blocks */
    char code_path[MAX_PATH_LENGTH]; /* Store the code path where the editor can jump to */
    commit_sha_type sha_id; /* A suitable data type to store a commit sha which can be opened */
    struct links node_links[MAX_LINKS]; /* Store the relevant links in this array */
}

struct idea{
    struct idea_node* head;
    int idea_id;
}

struct links{
    char link_address[200];
    char link_description[200];
}
```

Storage of user data

- MySQL.(don't know how to store data which keeps on changing but most probably most efficient)
- Plain text files.(easier but not efficient)
- Json files.(easier)
- Create custom objects like git blobs. Wrap content in magic numbers,(Need more info on this)

custom ncurses interface

- Need to implement basic houskeeping functions like:-

```
int draw_idea(struct idea* current_idea);
int save_idea(struct idea* current_idea);
int delete_idea(struct idea* current_idea);
int create_a_copy_of_idea(struct idea* current_idea);
```

- Need to implement flow chart editing functions like:-

```
int add_idea_block_to_the_right_of(struct idea_node* current_node);
int add_idea_block_to_the_top_of(struct idea_node* current_node);
int add_idea_block_to_the_bottom_of(struct idea_node* current_node);
int copy_this_block(struct idea_node* copy_source);
int paste_to_this_block(struct idea_node* paste_desitination);
int edit_this_block(struct idea_nore* current_node);
```

- Need to implement an idea window which will be a simple ncurses screen with keyboard enable navigation.
- Should be able to parse any given idea and draw a simple flowchart.
- No low level routines should be exposed to the main application.
- Press `c` for copy, `e` for edit, `v` for paste, and `q` for exit.
- Edit will open the idea node in an edit window which will show all parameters in a vim window and ready for edit.
- **Need to draw a state machine for this**.

Dir structure of `.flow` dir

- `.flowconfig` : config file.
  - Would need to write a small parser for parsing it.
  - Use `yacc` and `bison` for it.
  - It would be a good exercise to write a small parser in C.
  - Format of the configs would be simply `CONFIG_OPTION=<value>`.
  - Need to document all the config options in man page.
  - The default config file would have all options enabled if option is boolean otherwise a default value would be provided.
- Can store the `ideas` dir inside the `.flow` dir,
  - All `.json` flles would be stored inside it.

Controlflow of the application

- If launched without options then usage and help is printed.
- If launched with the option `idea_tracker` then the user is presented with home screen.
- Need to make a list of the required options. Will document it in the man page.

Home screen

- Welcome message is printed.
- If launched with proper options then it will look for a dir named `.flow` for config options.
- If `.flow` dir is not found then it will create it. for the will ask for 2 options i.e. what is the purpose research or code.
- Lets say research is selected.
- The control is transferred to the idea listing segment.

Idea listing

- Then it will look for a dir named `ideas` in the `.flow` dir and if it is not present then it will ask to create it.
- If the `ideas` dir is already present then it will display the oneline summary of all the ideas as a bullet list.
- Next it will give an option to select an idea number through a keyboard navigated selection.
- Once an idea is selected the program will pass on the control of idea files to the idea parser.

Idea parser

- It will check and parse the idea blobs and if try to pass on the information to the idea drawer screen.
- If no idea nodes are found then it will pass and empty parsed blob to initiate the drawing of the first block.

Idea drawer screen

- The idea drawer screen will try to draw the idea by using the parsed information.
- If empty blob recieved then it will draw the first block of the idea and pass on the control to the user to fill it.
- The user can save the idea here once filled and exit the drawer screen.
- The control will now be passed on to the idea listing segment.

Creating new idea

- By pressing `o` the user would be able to create a new idea. For simplicity I will keep the ideas sorted by their ids so the new idea would always be created at the end of the list.

Returning to last working idea

- An optional option when the application would be relaunched be to directly to go to the last idea the user was working on.

Man page

- Look into `scdoc` to generate man pages easily.