For specifying the maximum depth we use the `-maxdepth` level parameter. Similarly, we can also specify the minimum level at which the descending should start. If we want to start searching from the second level onwards, we can set the minimum depth using the `-mindepth` level parameter. To restrict the `find` command to descend to a maximum depth of 1, use the following command:

```
$ find . -maxdepth 1 -name "f*" -print
```

This command lists all the files whose names begin with "f", but only from the current directory. If there are subdirectories they are not printed or traversed. Similarly, `-maxdepth 2` traverses up to at most two descending levels of subdirectories.

`-mindepth` is similar to `-maxdepth`, but it sets the least depth level for the `find` traversal. It can be used to find and print the files that are located with a minimum level of depth from the base path. For example, to print all the files whose names begin with "f", and are at least two subdirectories distant from the current directory, use the following command:

```
$ find . -mindepth 2 -name "f*" -print
./dir1/dir2/file1
./dir3/dir4/f2
```

Even if there are files in the current directory or `dir1` and `dir3`, it will not be printed.

`-maxdepth` and `-mindepth` should be specified as the third argument to the `find` command. If they are specified as the fourth or further arguments, it may affect the efficiency of `find` as it has to do unnecessary checks (for example, if `-maxdepth` is specified as the fourth argument and `-type` as the third argument, the `find` command first finds out all the files having the specified `-type` and then finds all of the matched files having the specified depth. However, if the depth were specified as the third argument and `-type` as the fourth, `find` could collect all the files having at most the specified depth and then check for the file type, which is the most efficient way to search.

Search based on file type

Unix-like operating systems treat every object as a file. There are different kinds of files, such as regular file, directory, character devices, block devices, symlinks, hardlinks, sockets, FIFO, and so on.

The file search can be filtered out using the `-type` option. By using `-type`, we can specify to the `find` command that it should only match files having a specified type.

List only directories including descendants as follows:

```
$ find . -type d -print
```

It is hard to list directories and files separately. But `find` helps to do it. List only regular files as follows:

```
$ find . -type f -print
```

List only symbolic links as follows:

```
$ find . -type l -print
```

You can use the `type` arguments from the following table to properly match the required file type:

File type	Type argument
Regular file	`f`
Symbolic link	`l`
Directory	`d`
Character special device	`c`
Block device	`b`
Socket	`s`
FIFO	`p`

Search on file times

Unix/Linux filesystems have three types of timestamps on each file. They are as follows:

- **Access time** (`-atime`): It is the last timestamp of when the file was accessed by a user
- **Modification time** (`-mtime`): It is the last timestamp of when the file content was modified
- **Change time** (`-ctime`): It is the last timestamp of when the metadata for a file (such as permissions or ownership) was modified

There is no such thing as creation time in Unix.