

```

const assert = require('assert')
const cupreaddir = require('../index.js')
const fs = require('fs')
const path = require('path')

const temp = path.join(__dirname, 'temporary_test_folder')

before(function() {
  fs.mkdirSync(temp)
  // make 10 folders
  for (let i = 9; i >= 0; i--) {
    let subdir = path.join(temp, `${i}`)
    fs.mkdirSync(subdir)
    // place 10 files in each folder
    for (let j = 9; j >= 0; j--) {
      fs.writeFileSync(path.join(subdir, `${i}${j}.txt`), '\n')
    }
  }
})

describe('get all files from directory', function() {
  it('should return the correct number of files', function() {
    return cupreaddir.getAllFilePaths(temp).then((arr) => {
      assert.deepEqual(arr.length, 100)
    })
  })
})

describe('sort files in ascending order of creation time', function() {
  it('should produce a sorted array of fileInfo objects', async function() {
    let arr = await cupreaddir.getAllFilePaths(temp)
    let fileInfo = await cupreaddir.sort(arr, 'birthtimeMs')
    let fileNames = fileInfo.map((e) => {return path.basename(e.url)})
    let correctOrder = []
    for (let i = 9; i >= 0; i--) {
      for (let j = 9; j >= 0; j--) {
        correctOrder.push(`${i}${j}.txt`)
      }
    }
    assert.deepStrictEqual(fileNames, correctOrder)
  })
  describe('fileInfo object', function() {
    it('should return the filename', async function() {
      let arr = await cupreaddir.getAllFilePaths(temp)
      let fileInfo = await cupreaddir.sort(arr, 'birthtimeMs')
      assert.ok(typeof fileInfo[0].url === 'string')
    })
    it('should return the fs.Stat object', async function() {
      let arr = await cupreaddir.getAllFilePaths(temp)
      let fileInfo = await cupreaddir.sort(arr, 'birthtimeMs')
      assert.ok(typeof fileInfo[0].birthtime === 'object')
    })
  })
})

```

```
describe('sort files in descending order of creation time', function() {
  it('should produce a sorted array of fileInfo objects', async function() {
    let arr = await cupreaddir.getAllFilePaths(temp)
    let fileInfo = await cupreaddir.sort(arr, 'birthtimeMs', true)
    let fileNames = fileInfo.map((e) => {return path.basename(e.url)})
    let reverseOrder = []
    for (let i = 9; i >= 0; i--) {
      for (let j = 9; j >= 0; j--) {
        reverseOrder.push( `${i}${j}.txt` )
      }
    }
    assert.deepStrictEqual(fileNames, reverseOrder.reverse())
  })
})

after(function() {
  fs.rmdirSync(temp, {recursive: true})
})
```