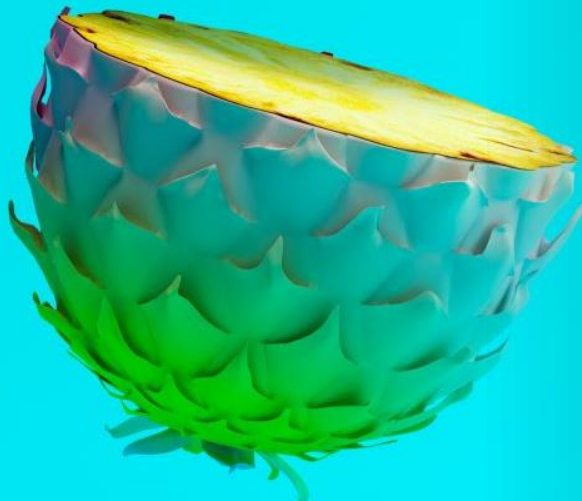




PROJECT UPDATE

Austin Lee, Jacob Mulroy, Conner
Hundt, D'Angelo Abell



Part 1: Create and update records

Successes:

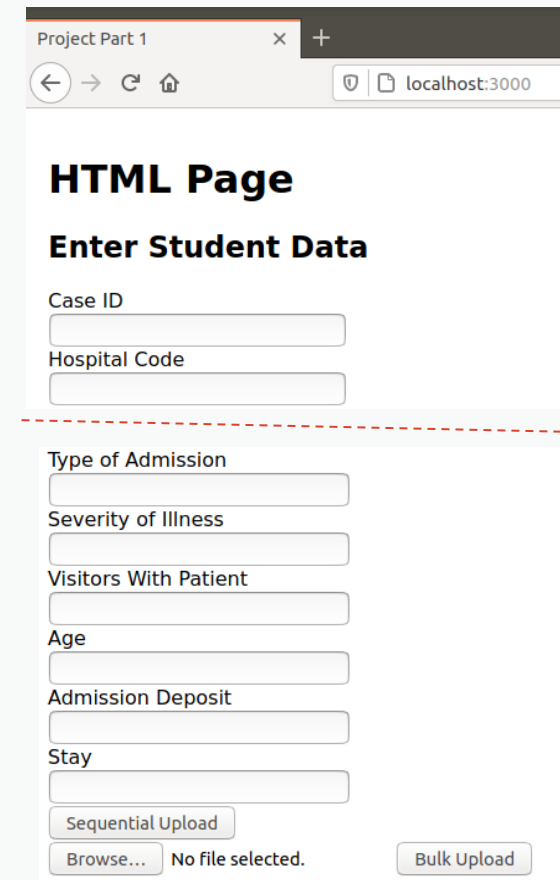
- Install/setup MongoDB

| Command | Function |
|------------------------------|-----------------------------------|
| Mongo | Init MongoDB service |
| use Students | Access existing database |
| db.dropDatabase() | Remove existing data from project |
| use Hospital | Create new project database |
| db.createCollection("info1") | Create collections for project |
| db.createCollection("info2") | |

Part 1: Create and update records

Successes:

- Modifying the HTML page
- Inserting files individually



Project Part 1 x +

localhost:3000

HTML Page

Enter Student Data

Case ID

Hospital Code

Type of Admission

Severity of Illness

Visitors With Patient

Age

Admission Deposit

Stay

Sequential Upload

Browse... No file selected. Bulk Upload

Part 1: Create and update records

Issues:

- **Unable to upload bulk through webpage**
- Add public folder and load via terminal

```
mongoimport --type csv -d Hospital -c info2 --headerline --drop DateSet.csv
```

- **Modifying variable length records**

```
db.info2.update({"Age":{"$eq: "21-30"}},  
{"$set":{"Patient Type":"Young"}},false, true)
```

Part 2: Sorting Algorithms

- Quicksort (python)
- Bubble sort (C#/python)
- MongoDB Sort

```
db.info.find({}).sort({"Age":1}).pretty()
```

```
db.info2.find().sort({"Age":1}).forEach(function(e)  
    {db.info2.insert(e);})
```

Part 2: Sorting Algorithms

- MongoDB issues

```
executionSuccess" : false,  
"errorMessage" : "Exec error resulting in state FAILURE :: caused by :: errmsg:  
\"Sort operation used more than the maximum 33554432 bytes of RAM. Add an index, or specify a  
smaller limit.\",
```

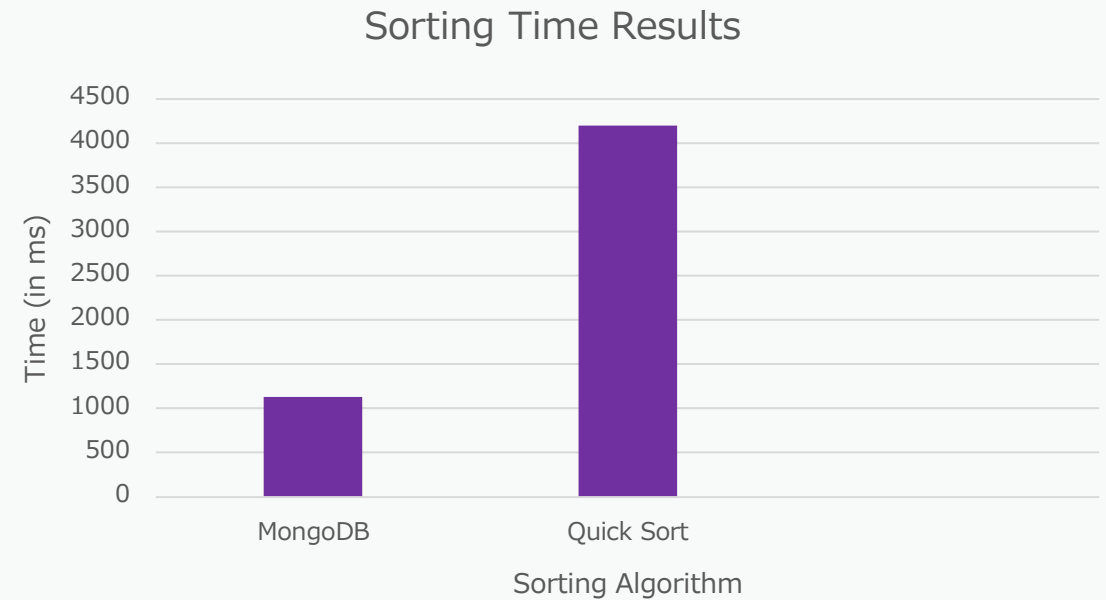
```
db.adminCommand({setParameter: 1, internalQueryExecMaxBlockingSortBytes:  
335544320})
```

Part 2: Sorting Algorithms

Average times (ms)

Quicksort: 4200

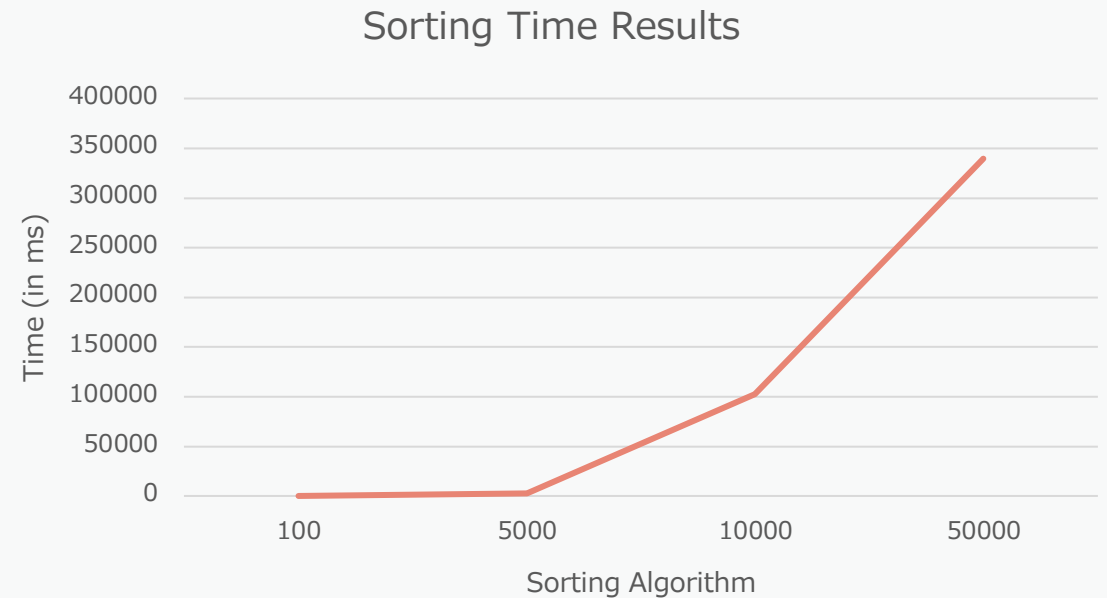
MongoDB: 1100



Part 2: Sorting Algorithms

Bubblesort:

| Number Records | Time (sec) |
|----------------|------------|
| 100 | 0.023 |
| 1000 | 0.226 |
| 5000 | 2.743 |
| 10000 | 10.251 |
| 50000 | 339.715 |
| 100000 | ??? |



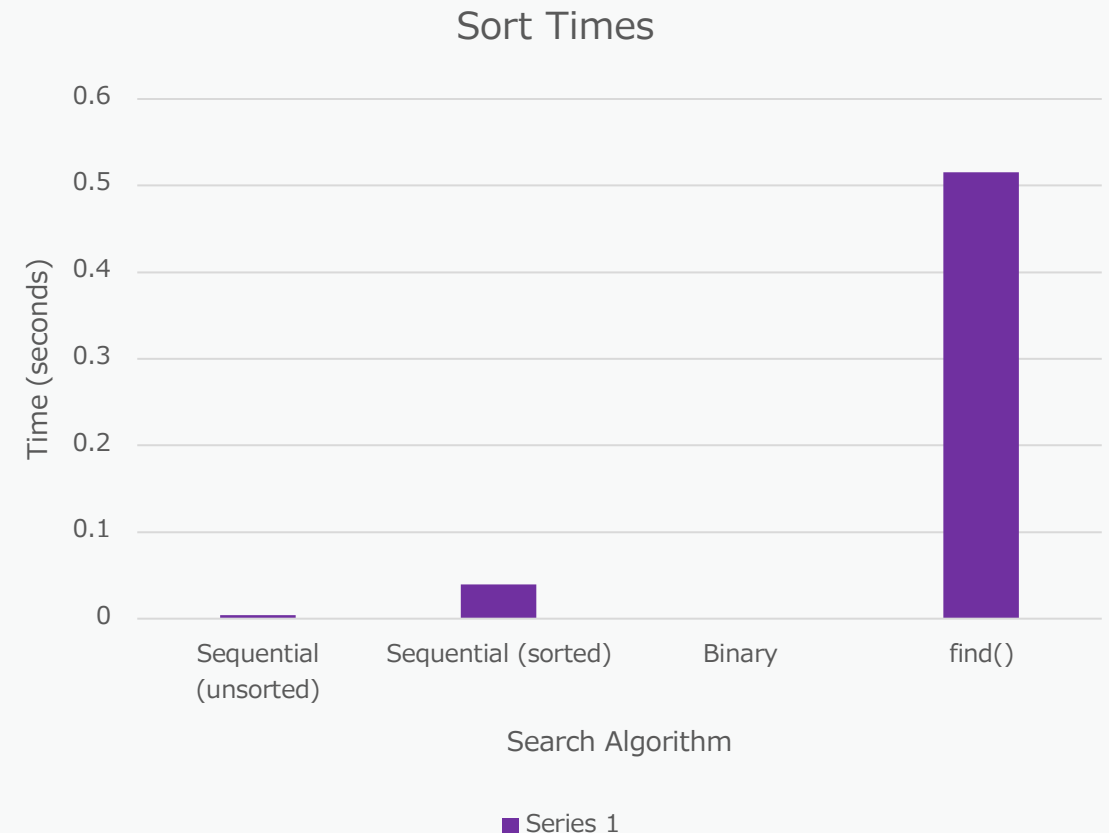
Part 3: Searching Algorithms

- Binary Search (python)
- Sequential Search (python)
- MongoDB Find

```
db.info2.findOne( {"Age": "21-30"} )
```

Part 3: Searching Algorithms

| Method | Time (sec) |
|----------------------|------------|
| Sequential(unsorted) | 0.004 |
| Sequential (sorted) | 0.04 |
| Binary | 0.000234 |
| find() | 0.515 |





FIN

