List of Commits

Refactor:

- Altering method return values to void (i.e changing boolean to void) … · EbenTowne/CSE464-EbenTowne@9a5b8d9
- Getting rid of unnecessary logic within the parseGraph method. The ad… · EbenTowne/CSE464-EbenTowne@514cea8
- Refactored addEdge operation to return boolean instead of int (easier… · EbenTowne/CSE464-EbenTowne@a284ebb
- Refactored DFS code to implement an actual queue rather than vectors … · EbenTowne/CSE464-EbenTowne@79e9a13
- Refactored BFS code to implement an actual queue rather than vectors … · EbenTowne/CSE464-EbenTowne@06ea079

Template Design Pattern:

- Saving first iteration of Template DESIgn Pattern · EbenTowne/CSE464-EbenTowne@7600e46
- Saving second iteration of Template Design Pattern (static issue from… · EbenTowne/CSE464-EbenTowne@eb9cd6d
- Better implementation of Template Design Pattern (more abstract metho… · EbenTowne/CSE464-EbenTowne@450247e

Strategy Design Patterns:

- First iteration of Strategy Design Pattern · EbenTowne/CSE464-EbenTowne@080a4aa
- First iteration of Strategy Design Pattern · EbenTowne/CSE464-EbenTowne@27a64cd

Random Traversal:

- First iteration of random walk algorithm · EbenTowne/CSE464-EbenTowne@185e9cd
- Merge pull request #5 from EbenTowne/master · EbenTowne/CSE464-EbenTowne@c3c9c82

Code Review Changes:

- Update DotGraph.java · EbenTowne/CSE464-EbenTowne@7daa060

Code Review comment:

```
41                      String dst = parts[1].trim();
42  -                   //System.out.println(src + " -> " + dst);
```

480jake 1 minute ago

Reformat the system output so the errors can be better recognized and easier to be debugged

Reply...

Resolve conversation

```
43  -                   if (!nodes.contains(src)) {
```

Code Output:

BFS/DFS Traversal:

```
Using BFS Strategy
Path Found: a->c->f

Using DFS Strategy
Path Found: a->d->g->h->f

Adding node: z

Added Edge: z -> a

Using BFS Strategy
Error: Path was not found between d and a!

Using DFS Strategy
Error: Path was not found between d and a!
```

Random Traversal:

```
random testing
Using Random Strategy
visiting Path{nodes=[Node{a}]}
visiting Path{nodes=[Node{a},Node{b}]}
visiting Path{nodes=[Node{a},Node{b},Node{c}]}
Path{nodes=[Node{a},Node{b},Node{c}]}

random testing
Using Random Strategy
visiting Path{nodes=[Node{a}]}
visiting Path{nodes=[Node{a},Node{b}]}
visiting Path{nodes=[Node{a},Node{b},Node{c}]}
Path{nodes=[Node{a},Node{b},Node{c}]}

random testing
Using Random Strategy
visiting Path{nodes=[Node{a}]}
visiting Path{nodes=[Node{a},Node{e}]}
visiting Path{nodes=[Node{a},Node{e},Node{g}]}
visiting Path{nodes=[Node{a},Node{e},Node{g},Node{h}]}
visiting Path{nodes=[Node{a}]}
visiting Path{nodes=[Node{a},Node{b}]}
visiting Path{nodes=[Node{a},Node{b},Node{c}]}
Path{nodes=[Node{a},Node{b},Node{c}]}

random testing
Using Random Strategy
visiting Path{nodes=[Node{a}]}
visiting Path{nodes=[Node{a},Node{e}]}
visiting Path{nodes=[Node{a},Node{e},Node{f}]}
visiting Path{nodes=[Node{a},Node{e},Node{f},Node{h}]}
visiting Path{nodes=[Node{a}]}
visiting Path{nodes=[Node{a},Node{b}]}
visiting Path{nodes=[Node{a},Node{b},Node{c}]}
Path{nodes=[Node{a},Node{b},Node{c}]}
```