

Aparajithan Venkateswaran

Recitation 13

TA: Yang Li

Instructor: Rhonda Hoenigman

Upon compiling and running for the first time, I encountered a segmentation fault:

```
$ g++ -std=c++11 Recitation13.cpp -o rec13
$ ./rec13
Segmentation fault (core dumped)
$
```

After running gdb, I found where the code broke and set a break point at that line to inspect code closely.

```
Program received signal SIGSEGV, Segmentation fault.
0x0000000000400e2e in Graph<int>::addVertex (this=0x7ffffc8032b0, n=1) at Recitation13.cpp:73
73         if(vertices[i].name == n){
(gdb)
```

```
(gdb) break 73
Breakpoint 1 at 0x400e19: file Recitation13.cpp, line 73.
(gdb) run
Starting program: /mnt/c/Users/Aparajithan/Dropbox/CSCI 2270/Recitations/rec-13/rec13
warning: Error disabling address space randomization: Success
warning: linux_ptrace_test_ret_to_nx: PTRACE_KILL waitpid returned -1: Interrupted system call

Breakpoint 1, Graph<int>::addVertex (this=0x7fffe8918110, n=1) at Recitation13.cpp:73
73         if(vertices[i].name == n){
(gdb) print i
$1 = 0
(gdb) print vertices.size()
$2 = 0
(gdb)
```

The program was trying to access the first element (index 0) in a vector when the vector had a size of 0. Upon inspecting the code, I found the following mistake in the end condition for the loop and corrected it.

```
72     for(int i = 0; i < vertices.size()+1; i++){
73         if(vertices[i].name == n){
74             found = true;
75             cout<<vertices[i].name<<" found."<<endl;
76         }
77     }
```

Now, fixing it and running the program resulted in an infinite loop:

[illegible]

Using gdb, I set a break point to where these print statements are called and inspected the loop variables.

```

89         for(int i = 0; i < vertices.size(); i++){
(gdb) next
90             cout<<vertices[i].name<<"-->";
(gdb) print i
$12 = 1
(gdb) next
91             for(int j = 0; j < vertices[i+1].adj.size(); j++){
(gdb) next
92                 i = j;
(gdb) next
93                 cout<<vertices[i].adj[j].v->name<<"***";
(gdb) print i
$13 = 0
(gdb) print j
$14 = 0
(gdb) next
91             for(int j = 0; j < vertices[i+1].adj.size(); j++){
(gdb) next
95                 cout<<endl;
(gdb) next
2-->2***

```

The `i = j` at line 92 keeps resetting `i` resulting in an infinite loop. And I removed that line in the code:

```

86     template<class T>
87     void Graph<T>::displayEdges(){
88         //loop through all vertices and adjacent vertices
89         for(int i = 0; i < vertices.size(); i++){
90             cout<<vertices[i].name<<"-->";
91             for(int j = 0; j < vertices[i+1].adj.size(); j++){
92                 i = j;
93                 cout<<vertices[i].adj[j].v->name<<"***";
94             }
95             cout<<endl;
96         }
97     }
98 }

```

After running the modified program, I ran into another segmentation fault.

```

$ g++ -std=c++11 -g Recitation13.cpp -o rec13
$ ./rec13
1-->2***
2-->4***
3-->
Segmentation fault (core dumped)
$

```

I set a break point and inspected the code in the debugger:

```
Program received signal SIGSEGV, Segmentation fault.
0x0000000000401094 in Graph<int>::displayEdges (this=0x7fffe5f3e480) at Recitation13.cpp:92
92      cout<<vertices[i].adj[j].v->name<<"***";
(gdb) b 92
Breakpoint 1 at 0x40106a: file Recitation13.cpp, line 92.
```

```
(gdb) next
91      for(int j = 0; j < vertices[i+1].adj.size(); j++){
(gdb) next
Breakpoint 1, Graph<int>::displayEdges (this=0x7fffe1c54f90) at Recitation13.cpp:92
92      cout<<vertices[i].adj[j].v->name<<"***";
(gdb) p i
$10 = 3
(gdb) p vertices.size()
$11 = 4
(gdb) p j
$12 = 0
(gdb) p vertices[i].adj.size()
$13 = 0
(gdb)
```

Once again, we are trying to access an element that is outside the size of the array and this caused due to line 91. So after fixing that and running the program, finally the program runs without errors:

```
$ g++ -std=c++11 -g Recitation13.cpp -o rec13
$ ./rec13
1-->2***
2-->4***
3-->4***
4-->
$
```