Clicker Questions

```
#include "stack.hpp"
#include <iostream>
#include <sstream>
#include <vector>
using namespace std;

void a(Stack *s)
{
  vector <Stack> v;

  v.resize(s->Size(), *s);    // Line F
  return;                     // Line G
}
```

```
int main()
{
  int n;
  int i;
  Stack s1;
  Stack s2;
  Stack *s3;
  ostringstream ss;

  cin >> n;
```

```
  for (i = 0; i < n; i++) {
    ss.clear();
    ss.str("");
    ss << i;
    s1.Push(ss.str());
  }
  s2 = s1;           // Line A
  s3 = new Stack;    // Line B
  *s3 = s2;          // Line C
  a(s3);             // Line D
  delete s3;         // Line E
  while (!s2.Empty()) cout << s2.Pop() << endl;
  return 0;
}
```

The answer to each of these will be *O(f(n))*. In your answer, just specify *f(n)*.

Question 1: What is the big-O running time of Line A?
Question 2: What is the big-O running time of Line B?
Question 3: What is the big-O running time of Line C?
Question 4: What is the big-O running from the beginning of Line D until just before Line F executes?
Question 5: What is the big-O running time of Line F?
Question 6: What is the big-O running from the beginning of Line G until just before Line E executes?
Question 7: What is the big-O running time of Line E?

Clicker Answers

- Question 1: This will call the assignment overload, and copy each of the *n* nodes from s1 to s2: *O(n)*.
- Question 2: This simply sets a pointer from the memory allocator. The pointer is to a few bytes, so this is *O(1)*.
- Question 3: This once again calls the assignment overload, which copies each of the *n* nodes from s2 to s3. *O(n)*.
- Question 4: This calls the procedure and copies the pointer, not the stack, as a parameter. This is *O(1)*.
- Question 5: The vector has *n* Stacks and each one is copied from a stack with *n* elements: $O(n^2)$.
- Question 6: Now, the destructor is called on each of those stacks, so it will delete n^2 Stacknodes: $O(n^2)$.
- Question 7: This calls the destructor on the stack, which deletes the *n* nodes: *O(n)*.