Product of array except itself

```
class Solution {
public:
    vector<int> productExceptSelf(vector<int>& nums) {
      int i;
      vector<int> awn;
      vector<int> before;
      vector<int> after;

      for (i = 0;i<nums.size();i++){
        before.push_back(0);
        after.push_back(0);
      }

      for (i = 0;i<nums.size();i++){
        if (i == 0){
          before[i] = 1;
        }else{
          before[i] = before[i - 1]*nums[i-1];
      }
      }

      for (i = nums.size()-1;i>=0;i--){
        if (i == nums.size() - 1){
          after[i] = 1;
        }else{
          after [i] = after[i+1]*nums[i+1];
        }
      }

      for (i = 0;i<nums.size();i++){
        awn.push_back(before[i]*after[i]);
      }

      return awn;
```

```
    }
};
```

- working solution
    - calculates the product before and after for each index and stores them in separate lists
    - then multiples the before and after to get the product of everything except itself