Shazaib Dawood
10/14/2024
CS 211

- Definition of recursion and example(s) of recursive function(s)
    - Recursion is where a function calls itself to complete an operation
    - base case terminates the function
    - recursive case calls function recursively
    - can have more than one base case or recursive case
    - example:
    - calculates the factorial by recursively calling the previous number and multiplying by it until the base case when 0 is reached

```
int factorial(n){
        if(n == 0){
        return 1;
        }
        else{
        return n * factorial(n - 1);
        }
}
```

- At least one recursive approach to producing the power set given an array of characters. Fully explain each part of the algorithm and how recursion is used.
    - power set is all possible subsets including the entire set and the null set
    - base case is the empty set
    - recursive case makes a power set for all elements except the first one, then for each power set generated create a new subset with the first element and without it
    - example: function powerSet() is recursively called twice in each step
        - includes current element and excluding the current element
        - current subset printed when n = 0

```
void powerSet(char set[], int n, char subset[], int subsetSize) {
   if (n == 0) {  // Base case: if set is empty, print the subset
     return;
   }

   // Exclude the current element
   powerSet(set + 1, n - 1, subset, subsetSize);

   // Include the current element
   subset[subsetSize] = set[0];
   powerSet(set + 1, n - 1, subset, subsetSize + 1);
```

}

- Definition of memoization and example(s) of how it can be used to optimize a recursive function (e.g. food web heights, Fibonacci sequence, factorial, etc.)
    - memoization stores the results of the function calls that use a lot of memory to reuse again when the inputs are the same. Helps minimize wasting memory for duplicate function calls
    - example: fibonacci(10) would store results for fibonacci(9) and fibonacci(8) so they don't need to be recalculated later

```
// Fibonacci function using memoization
int fibonacci(int n) {
   if (n <= 1) {
      return n;  // Base cases: fib(0) = 0, fib(1) = 1
   }

    // Compute and store the value in memo array
   memo[n] = fibonacci(n - 1) + fibonacci(n - 2);
   return memo[n];
}
```

- summary of each of the git commands listed in the Reading above.
    - config
        - configures git settings like username and email
        - repository effect: sets configurations for user identity so that commits to the repository are labeled back to the user correctly
        - git config --global user.name "Shazaib"
        - git config --global user.email "sdawo2@uic.edu"
    - clone
        - creates a local repository clone
        - repository effect: downloads the repository to the computer so you can work on it locally
        - git clone https://github.com/shazaib/my-repo.git
    - diff
        - shows the difference between 2 commits
        - repository effect: does not change the repository. Only shows local changes before they were committed
        - git diff
    - status
        - displays the state of the working directory and the staging area
        - repository effect: does not change the repository. Shows which files are staged, unstaged, and untracked

        - Git status
    - Add
        - stages changes for the next commit
        - repository effect: moves changes made in the working directory to the staging area
        - git add file.txt
    - rm
        - removes a file from the working directory and staging area
        - Repository effect: deletes the files and stages the removal for commit
        - git rm file.txt
    - commit -m
        - records staged changes to the repository with a message describing the commit
        - repository effect: creates a new commit with the current changes in the staging area
        - git commit -m “New Commit made”
    - Push
        - updates local commits to the main repository on github
        - repository effect: syncs local commits to the GitHub repository
        - git push origin main
    - pull
        - pulls changes from the main repository and merges them with the local branch
        - repository effect: updates the local branch to match the main branch
        - git pull origin main
    - log
        - shows a history of commits
        - repository effect: does not change the repository. Displays the commit history
        - git log