CSCI 1300

Switch, execution time, complexity analysis, auto, for range, sort, map and recursion

July 14th, 2021

University of Colorado Boulder

Be Boulder.

Please use the github link for the programing examples and slides.
https://github.com/rahul-aedula95/CSCI-1300

Switch case

- Alternatives for if else statements.
- Primarily used to make menus.
- Can only evaluate one variable at a time.
- Not as robust as if else statements but can reduce the amount of lines written for code.
- “default” is similar in purpose to the “else” statement in if and else.
- If you do not use break at the end of each switch you will execute all of the statements.

Auto and for-range

- Automatic type deduction in c++
- Can find the type based on the nature of the variable.
- Can only find the type during declaration not after!

- for-range allows looping over iterables without indexing.
- This means that you go through the items directly without needing to maintain position.
- This is only possible because of the “auto” type.

sort

- Handy function to sort if you do not have a direct implementation.
- called from the #include<algorithm> directive.
- syntax:
  - sort(start, end, comparingFunction);
- sorts it in ascending order by default if we only use the first two parameters
- We can use the third parameter (which is a function name) to decide the order in which we can sort items.

Maps

- located in the directive #include<map>
- Visualize it as a dictionary where each key has some value.
- All the keys are unique and all the values are not necessarily unique.
- Think of key as a unique index to some value.
- A fun experiment would be to try word count using map.

Measuring time for programs

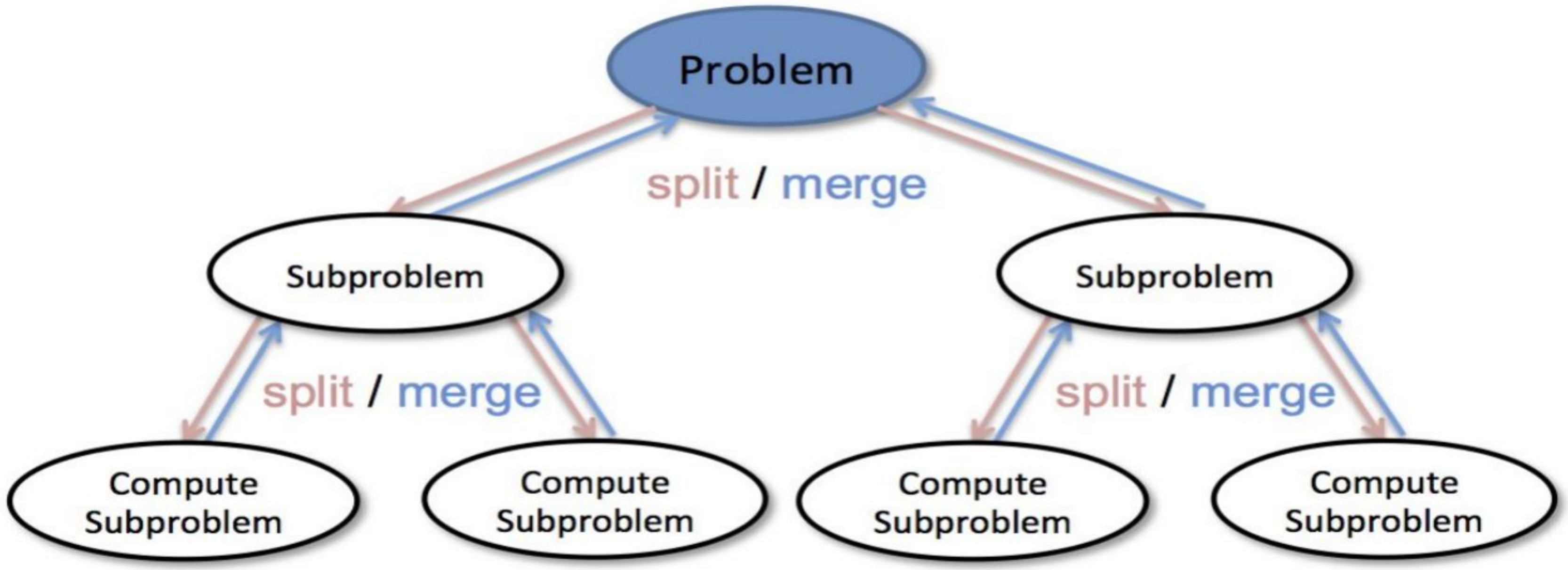
- Efficiency is a very important aspect of coding.
- Knowing how long a program runs can be vital to understand how to make it work faster.
- We utilize the functions present in the chrono library.
  - #include<chrono>
  - To find elapsed time we measure the time at the beginning and end of the code snippet.
  - We then subtract the ending time from the beginning .

Intuition of complexity measures

- We use the O (big oh) notation for stating what the complexity of a program is.
- This is a variable measure of how many steps a program takes to complete its task.
- The greater the measure the slower the program is.
- Let us know visualize a few:
  - O(1)
  - O(n)
  - O(n^2)

Recursion

Divide and Conquer



Recursion

1. Break into subproblems
2. Solve sub problems recursively
3. Have a base case to assemble the solution together.

- It is not the easiest solution to think of many times.
- Almost always you can solve it with some iterative method.
- Complexity should be nearly identical.