Recitation Class

Week 8

Homework

(e) $T(n) = 3T(n-2) + 5$ **(1 pts)**

$= 3^2 T(n-4) + 5 \cdot 3 + 5$

$= 3^3 T(n-6) + 5 \cdot 3^2 + 5 \cdot 3 + 5$

$= \dots$

$= 3^{\lfloor \frac{n}{2} \rfloor} T(n \bmod 2) + 5 \cdot 3^{\lfloor \frac{n}{2} \rfloor - 1} + 5 \cdot 3^{\lfloor \frac{n}{2} \rfloor - 2} + \dots + 5 \cdot 3^2 + 5 \cdot 3 + 5$

$= \frac{1}{2} \cdot 3^{\lfloor \frac{n}{2} \rfloor} - \frac{5}{2} = \Theta(3^{\lfloor \frac{n}{2} \rfloor})$

Homework

(f) $T(n) = 3T(n^{1/3}) + \Theta(\log n)$ **(1 pts)**

Let $x = \log n$ $\quad n = e^x$

$T(e^x) = 3T(e^{\frac{x}{3}}) + \Theta(x)$

let $S = T(e^x) = T(n)$

$\therefore S(x) = 3S(\frac{x}{3}) + \Theta(x)$

$\therefore S(x) = 3S(\frac{x}{3}) + \Theta(x) = 9S(\frac{x}{9}) + \Theta(x) + 3 \cdot \Theta(\frac{x}{3})$

$= \dots$

$= \Theta(x \log x)$

$\therefore T(n) = S(x) = \Theta(x \log x) = \Theta(\log n \log \log n)$

Homework

(bonus) $T(n) = T(n-1) + T(n-2)$ **(2 pts)**

Squeeze + guess & check method

1° lowerbound it by $T(n) \geq 2T(n-2)$

so we know $T(n) = \Omega(2^{\frac{n}{2}})$

2° upperbound it by $T(n) \leq 2T(n-1)$

so we know $T(n) = O(2^n)$

Hence $T(n) = 2^{\Theta(n)}$

More precisely, we know that runtime is exponential w.r.t n.

$\therefore T(n) = \Theta(\alpha^n)$

$\alpha^n = \alpha^{n-1} + \alpha^{n-2}$

$\therefore \alpha = \frac{1+\sqrt{5}}{2}$

$\because \alpha$ must be positive

$\therefore \alpha = \frac{1+\sqrt{5}}{2}$

$\therefore T(n) = \Theta((\frac{1+\sqrt{5}}{2})^n)$

Quiz

Q. 1 TRUE OR FALSE (14 pts)

1. Starting empty and doubling capacity only when full, there exists a sequence of m `push_back`s where one push costs $\Theta(m)$ while the average cost over all pushes is $O(1)$. ○ **T** ○ **F**

2. In a singly linked list with both head and tail pointers, `push_front`, `pop_front`, and `push_back` each run in worst-case $O(1)$ time. ○ **T** ○ **F**

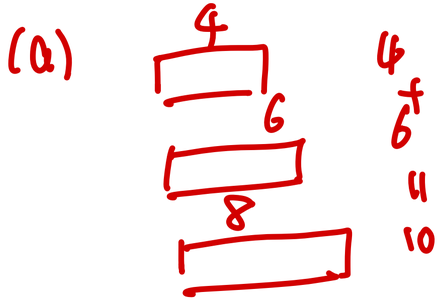
3. Given a pointer to a non-head and non-tail node in a singly linked list, deleting that node is a $\Theta(1)$ operation. ○ **T** ○ **F**

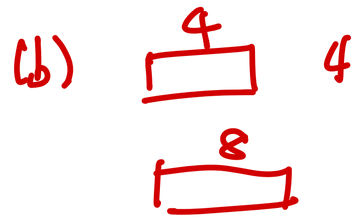
4. Evaluating the RPN expression `5 1 2 + 4 * + 3 -` yields 14. ○ **T** ○ **F**

5. With a hash table of size $m = 7$, hash $h(k) = k \bmod 7$, and *quadratic probing* $h_i(k) = (h(k) + i^2) \bmod 7$, inserting $10, 17, 24, 31$ occupies indices $3, 4, 0, 6$, and inserting 38 then succeeds at index 1. ○ **T** ○ **F**

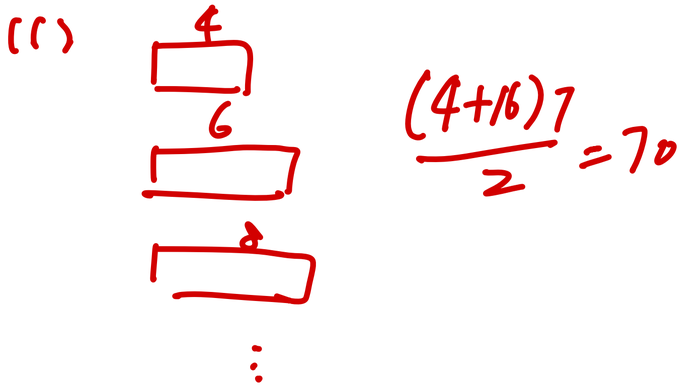
6. $n + \lfloor \log_2 n \rfloor = \Theta(n)$ and $(\log n)^2 = o(n^{0.01})$. ○ **T** ○ **F**

7. For fixed constants $A > B > 0$, $(\log n)^A = \omega((\log n)^B)$; moreover, for any $C > 0$, $(\log n)^C = o(n^\varepsilon)$ for every $\varepsilon > 0$. ○ **T** ○ **F**



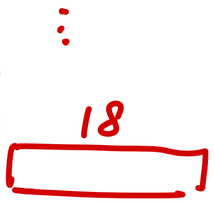


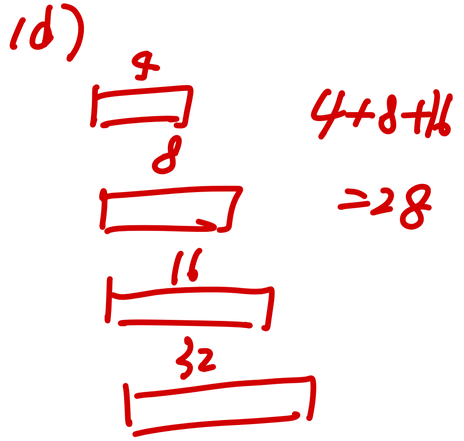
Quiz



Q. 2 ARRAY CAPACITY (8 pts)

Suppose there are two initially empty arrays of capacity 4. You continuously push elements into these arrays. When you want to push an element into a full array, you must increase the array's capacity and copy all the old elements to the new array. The first array's capacity increases by +2 each time. The second array's capacity increases by a factor of 2 each time. Answer the following questions; the subparts are independent.





(a) Suppose we insert 7 elements into the **first** array, the unused memory is 1, the total number of copies is 10.

(b) Suppose we insert 7 elements into the **second** array, the unused memory is 1, the total number of copies is 4.

(c) Suppose we insert 17 elements into the **first** array, the unused memory is 1, the total number of copies is 70.

(d) Suppose we insert 17 elements into the **second** array, the unused memory is 15, the total number of copies is 28.

Quiz

```
class MyStack {
  queue<int> q1, q2;
  void push(int x) {
    /*(1)*/;
    while (/*(2)*/) {
      int v = /*(3)*/;
      /*(4)*/;
    }
    std::swap(q1, q2);
  }
  int top() {
    /*(5)*/;
  }
  int pop() {
    int v = /*(6)*/;
    return v;
  }
  bool empty() const { return q1.empty(); }
};
```

Fill in the blanks here

(1) q2.push(x)

(2) !q1.empty()

(3) q1.pop()

(4) q2.push(v)

(5) return q1.front()

(6) q1.pop()

Quiz

Q. 4 ASYMPTOTIC ANALYSIS (10 pts)

```
inline void tiny_mix(int& x, int& y, int& z){
  int t = x ^ (y + 0x9e3779b9);
  x = y ^ (z + 0x7f4a7c15);
  y = z ^ (t + 0x85ebca6b);
}
void SolveB(vector<int>& a, int l, int r){
  int n = r - l + 1;
  if(n <= 1) return;
  int c1 = l + (3*n)/5;
  int c2 = l + (4*n)/5;
  int i1 = l, i2 = c1-1, i3 = c2-1;
  if(i1 >= l && i1 <= r && i2 >= l && i2 <= r && i3 >= l && i3 <= r){
    tiny_mix(a[i1], a[i2], a[i3]);
  }
  SolveB(a, l, c1-1);
  SolveB(a, l, c2-1);
}
```

$(l,\ l+\frac{3}{5}n-1) \longrightarrow \frac{3}{5}n$

$(l,\ l+\frac{4}{5}n-1) \longrightarrow \frac{4}{5}n$

First **write the correct recurrence** for the *number of calls* $T(n)$ made by `SolveB` on an input of size n, then **find a function** g such that $T(n) = \Theta(g(n))$. Show your reasoning; only the final $\Theta(\cdot)$ answer is worth 2 points.

$T(n) = T(\frac{3}{5}n) + T(\frac{4}{5}n)$

1° $T(n) \geq 2T(\frac{3}{5}n) \Rightarrow T(n) \geq \Theta(n^{\log_{\frac{5}{3}} 2})$

2° $T(n) \leq 2T(\frac{4}{5}n) \Rightarrow T(n) \leq \Theta(2^{\log_{\frac{5}{4}} n})$

$a^{\log_b c} = c^{\log_b a}$ $\leq \Theta(n^{\log_{\frac{5}{4}} 2})$

From these 2 inferences. we set $T(n) = \Theta(an^b)$

$\because T(1) = \Theta(1) \Rightarrow a = 1$

$\therefore T(n) = n^b$

$n^b = (\frac{3}{5})^b n^b + (\frac{4}{5})^b n^b \Rightarrow b \approx 2$

$\therefore T(n) = \Theta(n^2)$

CMake

Why Do I Need a Good Build System?

- You want to avoid hard-coding paths.
- You need to build the software on more than one machine.
- You must support multiple operating systems (even different Unix variants).
- You want to support multiple compilers.
- You prefer to describe your program's structure logically, not as a pile of flags and commands.
- You want to use third-party libraries.
- You'd like tools such as Clang-Tidy to assist your coding.
- You want to use a debugger effectively.
- You want to build and maintain a hug project.

CMake

Why Do I Need a Good Build System?

Name	Last commit message	Last commit date
..		
CMakeLists.txt	init commit	4 years ago
simple_example.cpp	init commit	4 years ago
simple_lib.cpp	init commit	4 years ago
simple_lib.hpp	init commit	4 years ago

Name	Last commit message	Last commit date
..		
apps	init commit	4 years ago
cmake	init commit	4 years ago
docs	init commit	4 years ago
include/modern	init commit	4 years ago
src	init commit	4 years ago
tests	init commit	4 years ago
.gitignore	init commit	4 years ago
CMakeLists.txt	init commit	4 years ago
README.md	init commit	4 years ago

CMake

Building a project

- Unless otherwise noted, you should always make a build directory and build from there.

```
~/package $ mkdir build
~/package $ cd build
~/package/build $ cmake ..
~/package/build $ make
```

mkdir — create a new directory

cd — move into (change to) a directory

.. — the parent (one level up) directory

. — the current directory

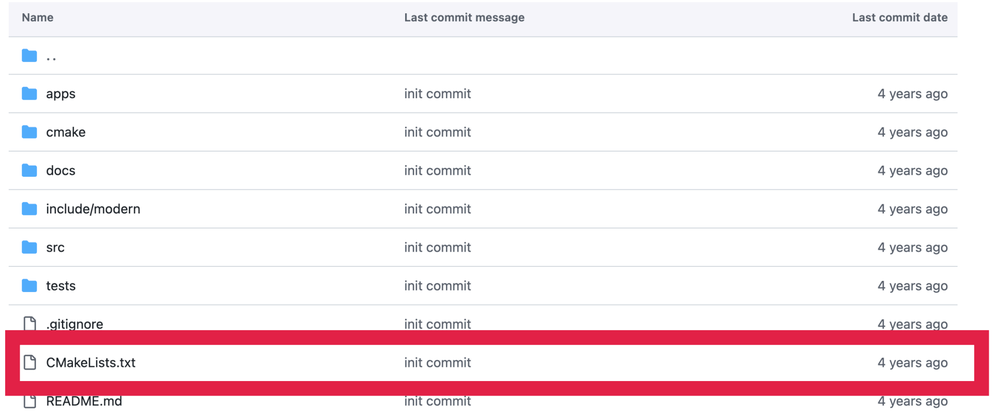
CMake

Standard options

- `-DCMAKE_BUILD_TYPE=`
  - Pick from Release, RelWithDebInfo, Debug, or sometimes more.
- `-DCMAKE_INSTALL_PREFIX=`
  - The location to install to. System install on UNIX would often be `/usr/local` (the default), user directories are often ~/.local, or you can pick a folder.
- `-DBUILD_SHARED_LIBS=`
  - You can set this `ON` or `OFF` to control the default for shared libraries (the author can pick one vs. the other explicitly instead of using the default, though)
- `-DBUILD_TESTING=`
  - This is a common name for enabling tests, not all packages use it, though, sometimes with good reason.

CMake

CMakeLists.txt



Name	Last commit message	Last commit date
..		
apps	init commit	4 years ago
cmake	init commit	4 years ago
docs	init commit	4 years ago
include/modern	init commit	4 years ago
src	init commit	4 years ago
tests	init commit	4 years ago
.gitignore	init commit	4 years ago
CMakeLists.txt	init commit	4 years ago
README.md	init commit	4 years ago

- See if you can follow the following file.
- It makes a simple C++11 library and a program using it.

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

Minimum Version

- Here's the first line of every CMakeLists.txt
- which is the required name of the file CMake looks for

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

Setting a project

```
project(MyProject VERSION 1.0
                  DESCRIPTION "Very nice project"
                  LANGUAGES CXX)
```

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

Making an executable

```
add_executable(one two.cpp three.h)
```

- Creates an executable target named one.
- Compiles two.cpp.
- Ignores three.h for compilation.

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

Making a library

```
add_library(one STATIC two.cpp three.h)
```

- Purpose: Create a library target named one.
- Types: STATIC (static), SHARED (dynamic).

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

Making a Library

```
target_include_directories(one PUBLIC include)
```

- adds an include directory to a target.

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

Making a Library

```
add_library(another STATIC another.cpp another.h)
target_link_libraries(another PUBLIC one)
```

- adds an include directory to a target.

```
cmake_minimum_required(VERSION 3.15...4.0)

project(Calculator LANGUAGES CXX)

add_library(calclib STATIC src/calclib.cpp include/calc/lib.hpp)
target_include_directories(calclib PUBLIC include)
target_compile_features(calclib PUBLIC cxx_std_11)

add_executable(calc apps/calc.cpp)
target_link_libraries(calc PUBLIC calclib)
```

CMake

A simple example

Name	Last commit message	Last commit date
..		
CMakeLists.txt	init commit	4 years ago
simple_example.cpp	init commit	4 years ago
simple_lib.cpp	init commit	4 years ago
simple_lib.hpp	init commit	4 years ago

```
cmake_minimum_required(VERSION 3.1...3.21)

project(
    ModernCMakeExample
    VERSION 1.0
    LANGUAGES CXX)

add_library(MyLibExample simple_lib.cpp simple_lib.hpp)
add_executable(MyExample simple_example.cpp)
target_link_libraries(MyExample PRIVATE MyLibExample)
```

```
~/package $ mkdir build
~/package $ cd build
~/package/build $ cmake ..
~/package/build $ make
```

CMake

A simple example

Name	Last commit message	Last commit date
..		
apps	init commit	4 years ago
cmake	init commit	4 years ago
docs	init commit	4 years ago
include/modern	init commit	4 years ago
src	init commit	4 years ago
tests	init commit	4 years ago
.gitignore	init commit	4 years ago
CMakeLists.txt	init commit	4 years ago
README.md	init commit	4 years ago

```
~/package $ mkdir build
~/package $ cd build
~/package/build $ cmake ..
~/package/build $ make
```

How does the CMakeLists.txt Works?