Cheatography

Advanced Programming with C++ Cheat Sheet

by Jonathan_Walsh1999 via cheatography.com/81859/cs/19567/

Stacks

Way of sorting data

Data Structure

Last in First Out (LIFO)

Push Back - Add item to top of stack

Pop Back - Remove item from top of stack

Big O

Expresses the efficiency for an algorithm

Can express worst, average or best case

Mathematical analysis

Measure of computational resources

Example: O(n^2) takes longer than O(2n)

Logarithms

Base 2 is assumed (NOT 10)

So logn is log*2*n

log*2*n = a

is Equivalent to

2^a = n

Linked Lists

Object is held in a black of memory

An instance of a class includes a pointer to another instance of the same class

[]-> []-> []->

Last in First Out (LIFO)

2 Classes are used

The end of the list is 0

Example:

Push Front method

1. 'Head' points at 0
2. Create a *new* element
3. Set 'next' pointer of *new* element to point at head
4. Set 'head' to point at *new* element

Stack

Value objects are pushed onto the stack.

Examples:

```
int i = 4;
float j = 3.4f;
double k = 7.8;
char l = 'h';
```

Pointers

Quick Sort

Very Efficient Algorithm

Uses Recurision

Big O Notation - O(nlogn)

Algorithm:
- If array is NOT empty
- Choose a number is act as a pivot (usually rightmost element)
- Partition the array into 2 sections
- If a number is smaller than pivot move into first section
- Else move into second section
- Recurse for both sections until you have one element left

Recursion

A Function that calls itself

Hard to debug

Better to use loop in most cases

Process:
- Function Call - carry out operations
- Place state of function onto stack
- Call Recurse function - carry out operations
- Place state of recurse function onto stack
- Call the recurse of the recurse function etc.

Templates

Generic Programming

```
template <classT> void trivial(T term)
```

`T` can be any data type

Casting

Change the data type of a variable to another

Examples:

```
float var1 = 34.8f;
int var2 = static_cast<int>(var1);
int i = 4;
char ptr = reinterpret_cast<char>(&i);
```

Heap

Object created with the `new` keyword are pushed onto the heap

Deep Copy

Copy of object and any dynamically allocated memory pointed to by the object

?

a > b ? max = a : max = b;

Is equivalent to

if (a > b) max = a; else max = b;

Hash Table

Array type Data Structure

Fast storage/retrieval method

Generates integer value

Example: Turns name into a number which is the index of the array

Big O Notation - O(1) - Very Fast

Hashing Methods

Linear Probing

Search for an empty space using linear search

Quadratic Probing

Searching for an empty space making larger steps

e.g. 1, 4, 9, 16... rather than 1,2,3,4... with linear probing

Chained Hash Table

Each location is head of chain

Each location has associated linked list

Retrieval means chain must be searched

Each location is effectively a bucket

Open Address Hash Table

All keys are stored directly in the table

Everything gets stored in table

Collisions resolved by probing

Avoid making the hash table larger than 80% capacity to avoid collisions

Bitwise

Changing the binary bits in a variable

Example:

```
x = x << 1; //Shift bits left by 1
```

12 is in binary 0000 1100

Shifting left by 1 changes it to

0001 1000 which is 24

Variables which point to a space in memory

Example:

```
int* ptr = new int;
delete ptr;
```

Inline

Function which is saved in memory

Usually used for smaller helper functions

Speeds up compile time but results in more memory being used up

Shallow Copy

Copy of Object but not any dynamically allocated memory pointed to by the object

Function Pointers

Used for pointing at function calls

Example:

```
#include <functional>
int foo(float a, float b);
std::function<int(float,float)>
fp; //points at function which
returns an int and has 2 float
parameters
fp = &foo;
```

By **Jonathan_Walsh1999**
cheatography.com/jonathan-walsh1999/

Not published yet.
Last updated 6th May, 2019.