Lecture 1

Introduction

Dr. Yusuf H. Sahin
Istanbul Technical University

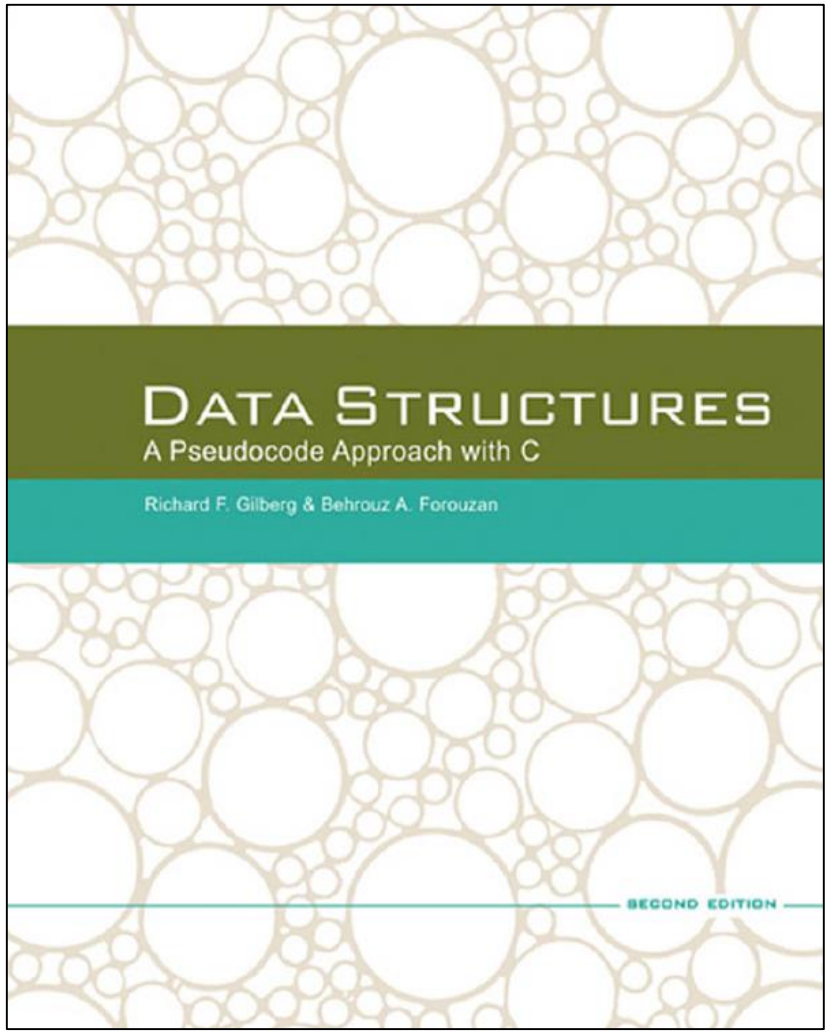
sahinyu@itu.edu.tr

Before starting...

- Plagiarism: What is Prohibited?
  - Using published or online sources directly
  - Copy-pasting
  - Writing reports without paraphrasing

- Course Book:

  Data Structures: A Pseudocode Approach with C, Second Edition
  Richard F. Gilberg & Behrouz A. Forouzan, Thomson Learning.



Pseudocode

- Pseudocode is an English-like representation of algorithm logic.
  - Combines plain language with structured code elements.
  - Helps to outline the steps of an algorithm before writing the actual code.

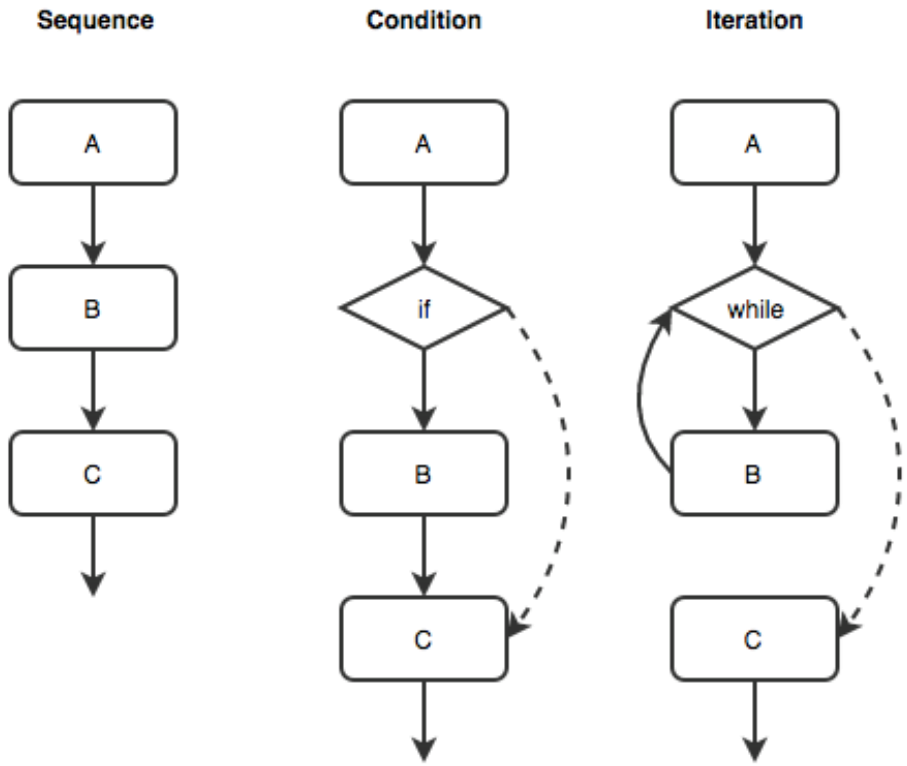
- Each algorithm begins with a header:
  - Name of the algorithm.
  - Parameters required by the algorithm.
  - Describes any preconditions (initial conditions) and postconditions (expected outcomes).

```
Algorithm Search(list, argument, location)
Search array for specific item and return index location.

Pre list contains data array to be searched argument
contains data to be located in list
Post location contains matching index -or- undetermined if
not found
Return true if found, false if not found

    Initialize found as false
    Set location to -1  // Default value if not found

    For each index i from 0 to length of list - 1 do:
        If list[i] equals argument then:
            Set location to i
            Set found to true
            Exit loop

    If found then:
        Return true  // Item found, location contains index
    Else:
        Return false  // Item not found, location remains -1
End Algorithm
```

Structural & Procedural Programming

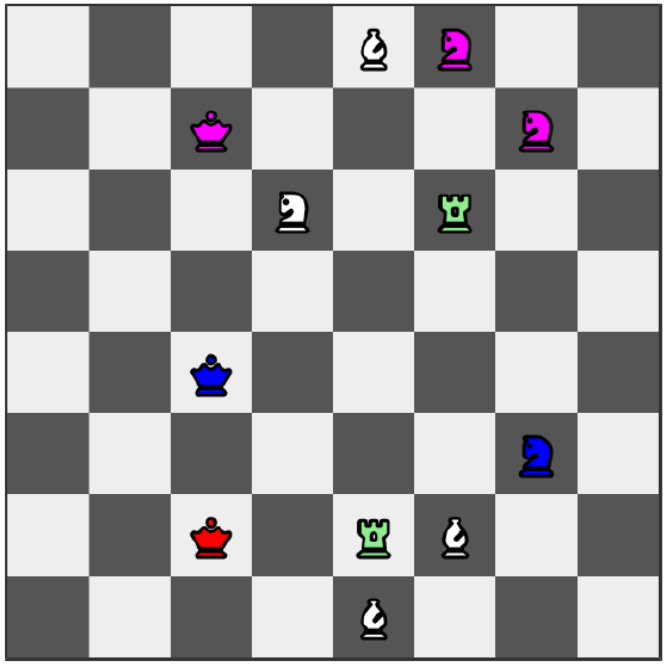
- **Structured Programming**
  - Böhm–Jacopini theorem: Any computable function can be implemented using three «structures»: Sequence, condition, iteration
  - The term is dedicated to Edsger Wybe Dijkstra because of his work «Notes on Structured Programming»

- **Procedural Programming**
  - The program is divided into procedures (functions).
  - C is a pure procedural programming language.



What is a data structure?

A chessboard containing pieces of multiple types & teams. How could I store them?



- A particular way to organize the data
  - How to «structure» the information
- Why we need to organize the data?
  - Create, insert, delete, display, search, sort etc.
- Built-in data types are primitive data structures: Int, Float, Char, Bool
- An array of a built-in data type is the most basic non-primitive data structure.

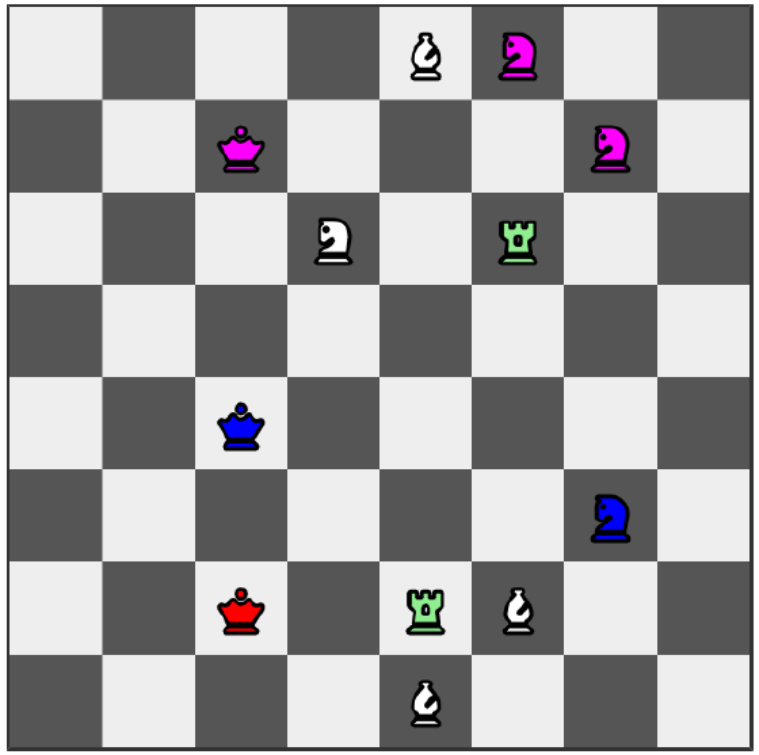
```c
#include <stdio.h>

int main() {
    int primes[14] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43};

    for (int i = 0; i < 14; i++) {
        printf("%d ", primes[i]);
    }

    return 0;
}
```

primes

2	3	5	7	11	13	17	19	…

What is a data structure?

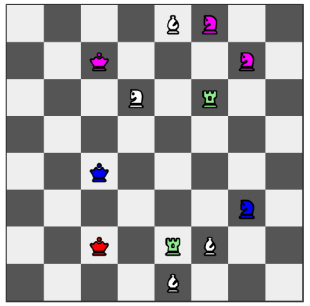
- The most basic representation is not applicable generally.



```
char piece[8][8] = {
    {'e','e','e','e','b','k','e','e'},
    {'e','e','q','e','e','e','k','e'},
    // Fill in the rest of the array as needed
};

char team[8][8] = {
    {'-','-','-','-','w','p','-','-'},
    {'-','-','p','-','-','-','-','-'},
    // Fill in the rest of the array as needed
};

printf("%c\n", piece[0][5]);
```

- An effective data structure should be selected to represent the given problem.

`chessboard my_board =`



Basic Terms

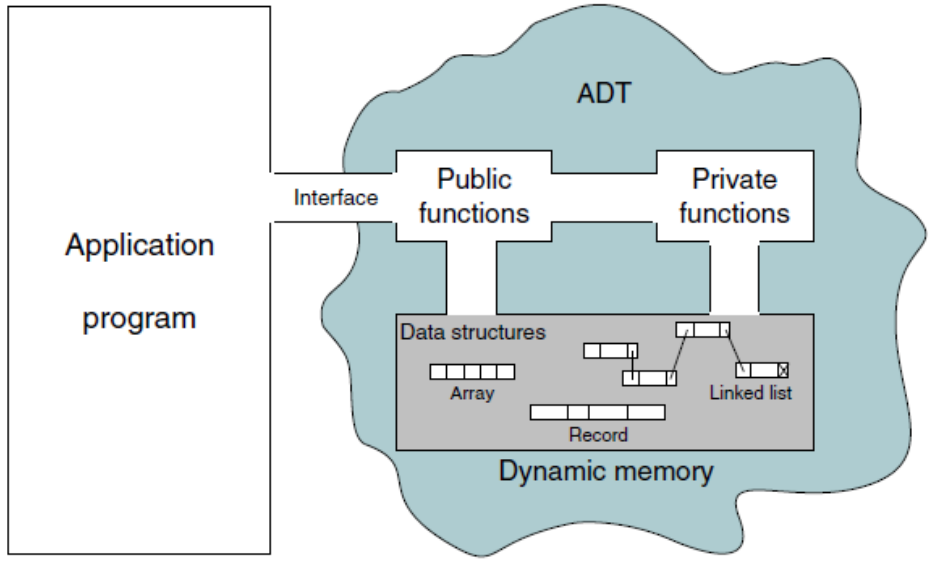
- **Atomic Data**: Data that is indivisible and represents a single piece of information.
- **Composite Data**: Data that can be divided into meaningful subfields.
- **Data Type:** A combination of a data set (e.g. 32 bit integers) and the operations (e.g. +,-,*,/,…) that can be applied to it.
- **Data Structure:** A collection of atomic and composite data with defined relationships.

```c
#include <stdio.h>

// Define an enum for the piece types
typedef enum {
    EMPTY, PAWN, ROOK, KNIGHT, BISHOP, QUEEN, KING
} PieceType;

// Define an enum for the piece colors
typedef enum {
    NONE, WHITE, BLACK, RED, BLUE
} PieceColor;

// Define a structure to represent a chess piece
typedef struct {
    PieceType type;
    PieceColor color;
} ChessPiece;
```

```c
// Define the chess board as an 8x8 array of ChessPiece structures
ChessPiece board[8][8];

int main() {
    board[0][4].type = KING;
    board[0][4].color = WHITE;

    board[7][0].type = ROOK;
    board[7][0].color = BLACK;

    int kingcount = 0;

    for(int i=0;i<8; i++)
        for(int j=0;j<8; j++)
            if(board[i][j].type == KING)
                kingcount++;

    printf("King count: %d\n", kingcount);

    return 0;
}
```

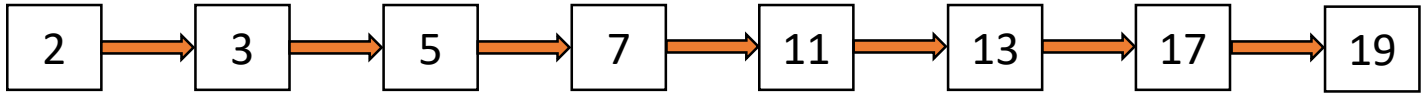
Abstract Data Type

- An Abstract Data Type is a concept that defines a data structure purely in terms of its behavior (what it can do) rather than its implementation (how it does it).
- ADTs define functions for use while hiding the implementation details.
- Abstraction generalizes operations, leaving the specifics of implementation hidden.
- In an ADT, users interact with data through operations like insert and retrieve, without knowing or needing to know the underlying structure.

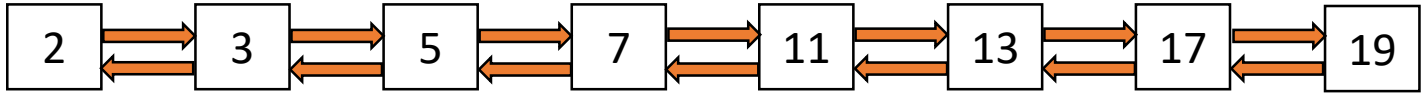


Foreshadowing: Linear Data Structures

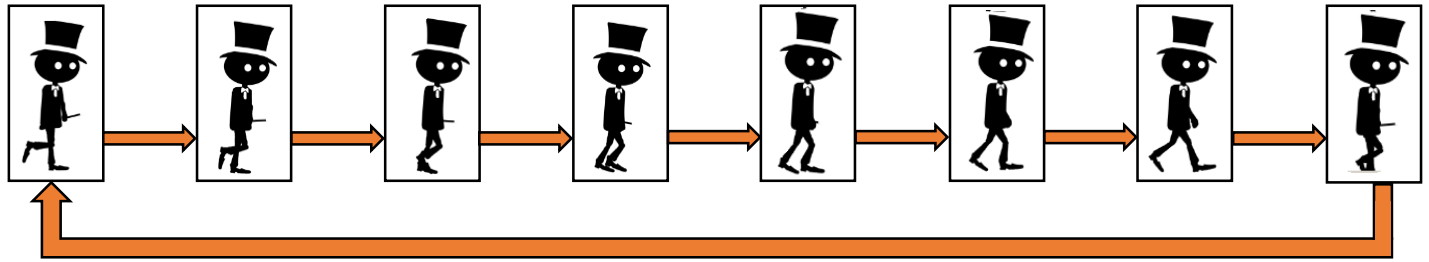
- Linked List
  - From each element, the next element could be reached.



- Double Linked List
  - From each element, the next and previous elements could be reached.



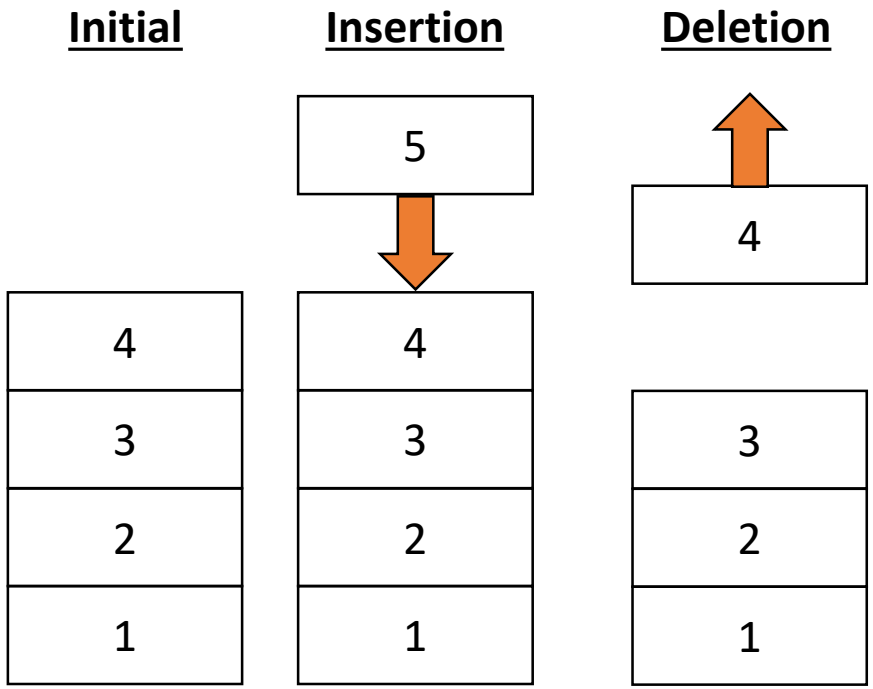
- Circular Linked List
  - From the last element, the first element could be reached.



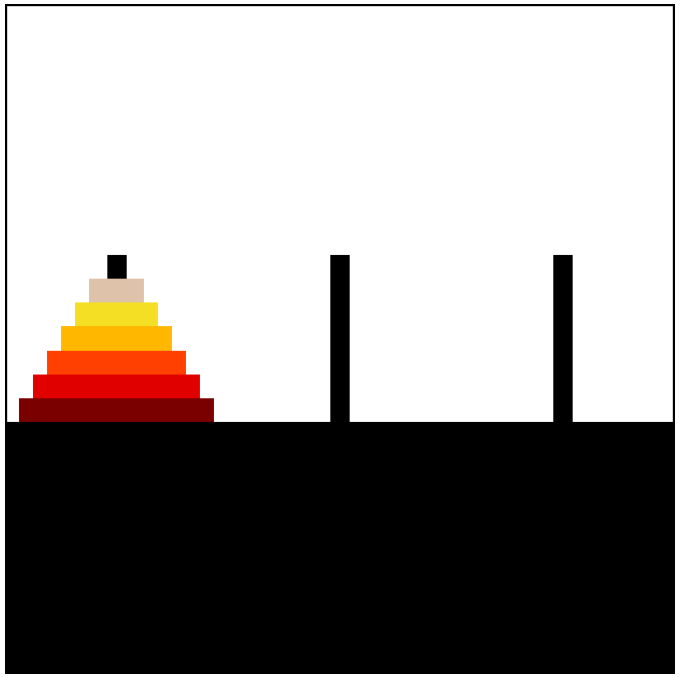
https://www.youtube.com/watch?v=2j4FfSfJ4V0

Foreshadowing: Linear Data Structures

- Stack
  - Add new element on the top
  - Delete the element on the top



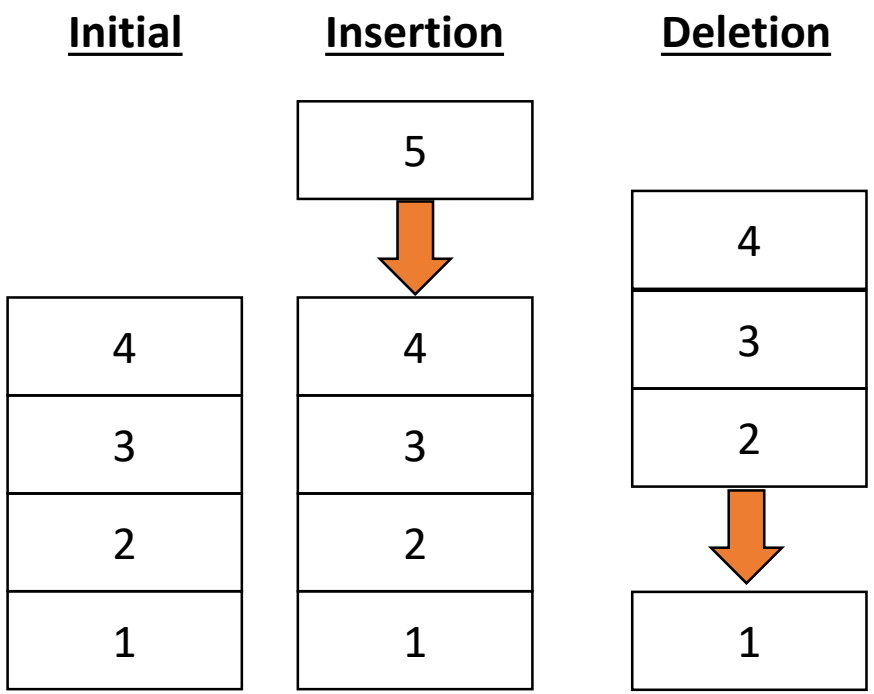
The Towers of Hanoi

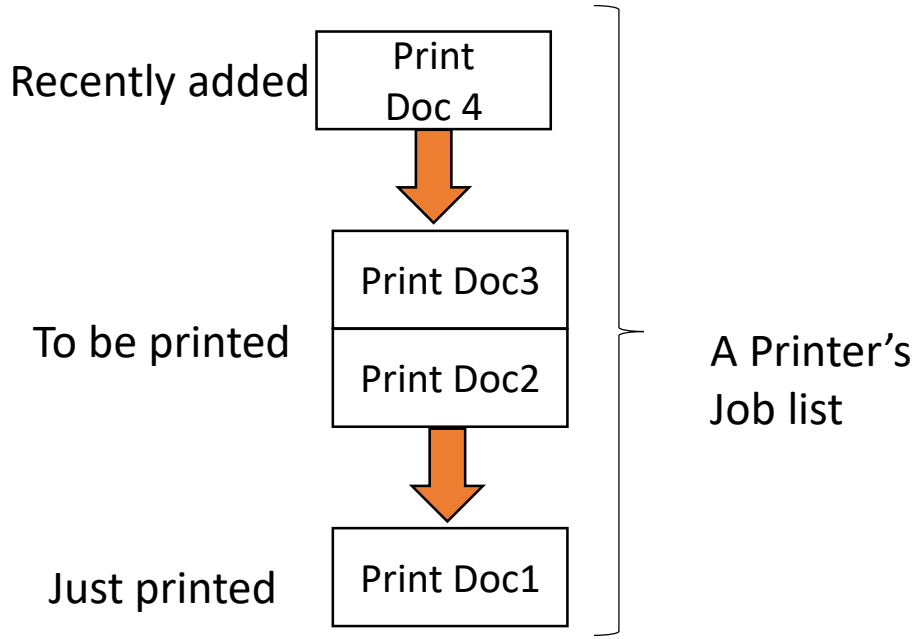


https://en.wikipedia.org/wiki/Tower_of_Hanoi#/media/File:Iterative_algorithm_solving_a_6_disks_Tower_of_Hanoi.gif

Foreshadowing: Linear Data Structures

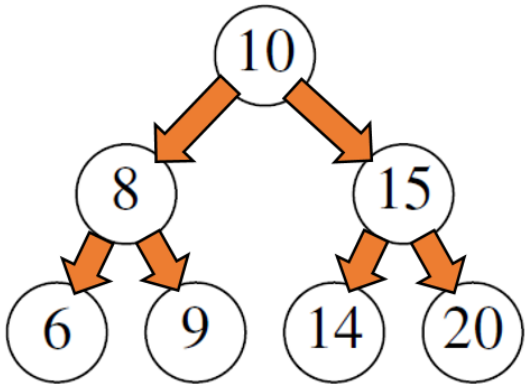
- Queue
  - Add new element to the back
  - Delete the element at the front



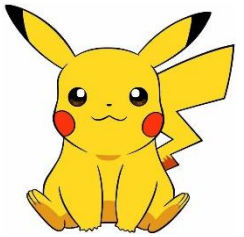


Foreshadowing: Trees

- Hierarchical data structures
- Can be binary
- or n-ary



A pokemon match



vs

Level 0
(Initial position)

PIKACHU	HP	273
	PP	100
BLASTOISE	HP	361
	PP	100
STATS	TURN	P

Level 1
(Pikachu's first attack)

PIKACHU	HP	273
	PP	90
BLASTOISE	HP	321
	PP	100
STATS	TURN	B
Pikachu used Thundershock. It's effective.		

PIKACHU	HP	273
	PP	85
BLASTOISE	HP	311
	PP	100
STATS	TURN	B
Pikachu used Skull Bash. It's effective.		

PIKACHU	HP	273
	PP	85
BLASTOISE	HP	361
	PP	100
STATS	TURN	B
Pikachu used Skull Bash. It's not effective.		

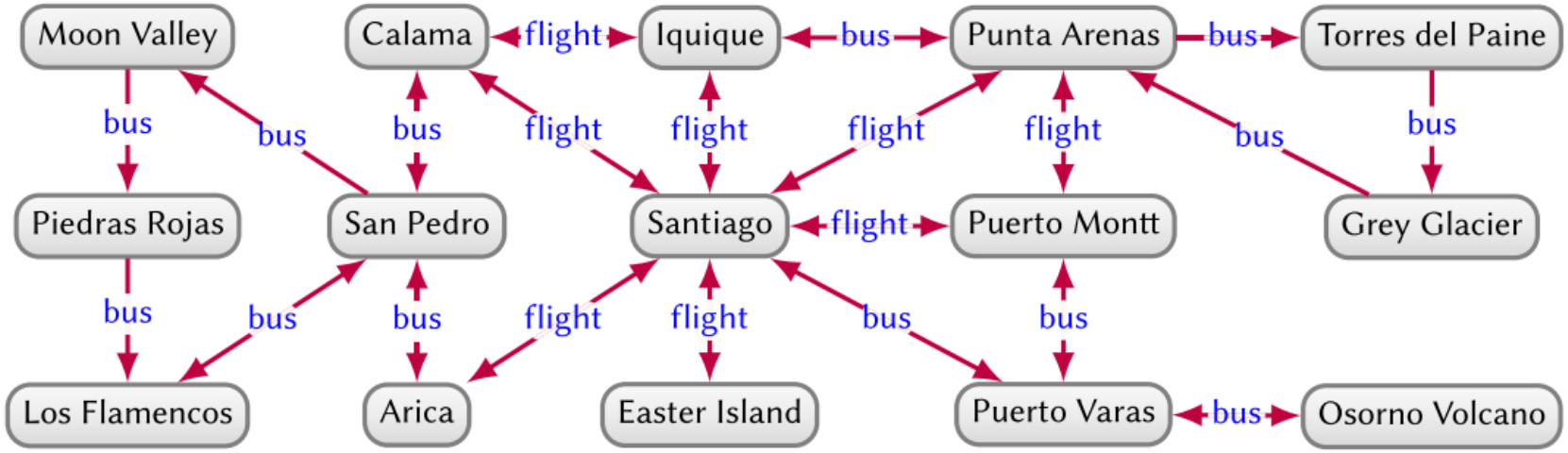
PIKACHU	HP	273
	PP	80
BLASTOISE	HP	301
	PP	100
STATS	TURN	B
Pikachu used Slam. It's effective.		

PIKACHU	HP	273
	PP	80
BLASTOISE	HP	361
	PP	100
STATS	TURN	B
Pikachu used Slam. It's not effective.		

Level 2
(will continue…)

Foreshadowing: Graphs

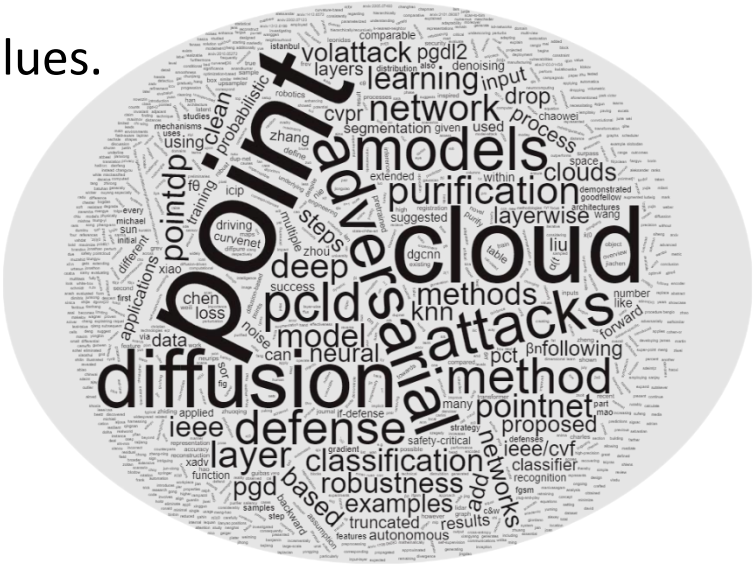
- A data structure where the elements (nodes) are connected via edges.
- Could be directed or undirected.
- Edges could also contain a property.



Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., ... & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys (Csur)*, *54*(4), 1-37.

Foreshadowing: Sets & Dictionaries

- A set is a container that stores a collection of unique values.
  - **Ex:** Set of all words from an article



- A dictionary is a container that keeps associations between *keys* and *values*.
  - **Ex:** The mapping of words from the article to their usage counts.