CSCE 4263
Advanced Data Structures

Lecture 2

Review

Fall 2025

Prof. Khoa Luu
khoaluu@uark.edu

Announcements

- Course website has been updated.
- **Assignment 1 – Available this Week**
- Review background

Pointers

Pointers

```
int  x = 5;
```

Pointers

```
int  x = 5;
int  *p = &x;
```

Pointers

```
int  x = 5;
int  *p = &x;
int  k = *p;
```

Pointers

```
int  x = 5;
int  *p = &x;
int  k = *p;
printf("%d\n", p);
```

Pointers

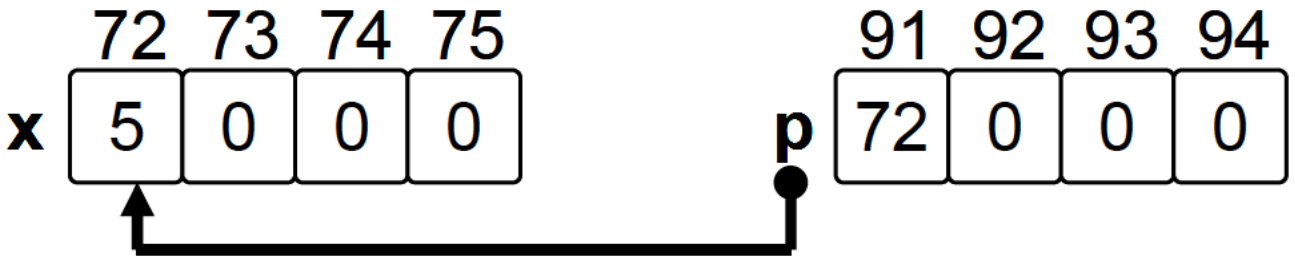
```
int  x = 5;
int  *p = &x;
int  k = *p;
printf("%d\n", p);
printf("%d\n", *p);
```

Pointers

```
int  x = 5;
int  *p = &x;
int  k = *p;
printf("%d\n", p);
printf("%d\n", *p);
printf("%d\n", &p);
```

Pointers

```
int  x = 5;
int  *p = &x;
int  k = *p;                    // get value of x
printf("%d\n", p);              // get address of x
printf("%d\n", *p);             // get value of x
printf("%d\n", &p);             // get address of p
```

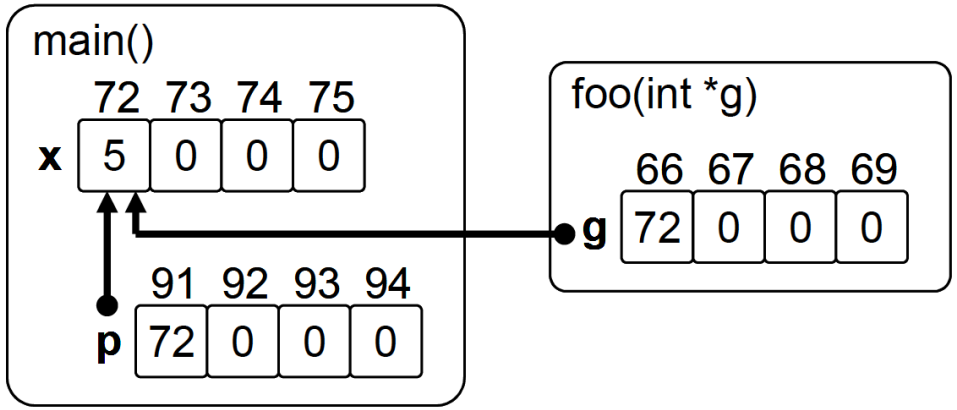


Pointers

```
void foo(int *g)
{
        *g = *g + 1;
        g = g + 1;
}

void main()
{
        int  x = 5;
        int  *p = &x;

        foo(p);
        foo(&x);

}
```

Pointers

```
void foo(int *g)
{
        *g = *g + 1;
        g = g + 1;
}

void main()
{
        int  x = 5;
        int  *p = &x;

        foo(p);
        foo(&x);
         // value of x has changed
}
```

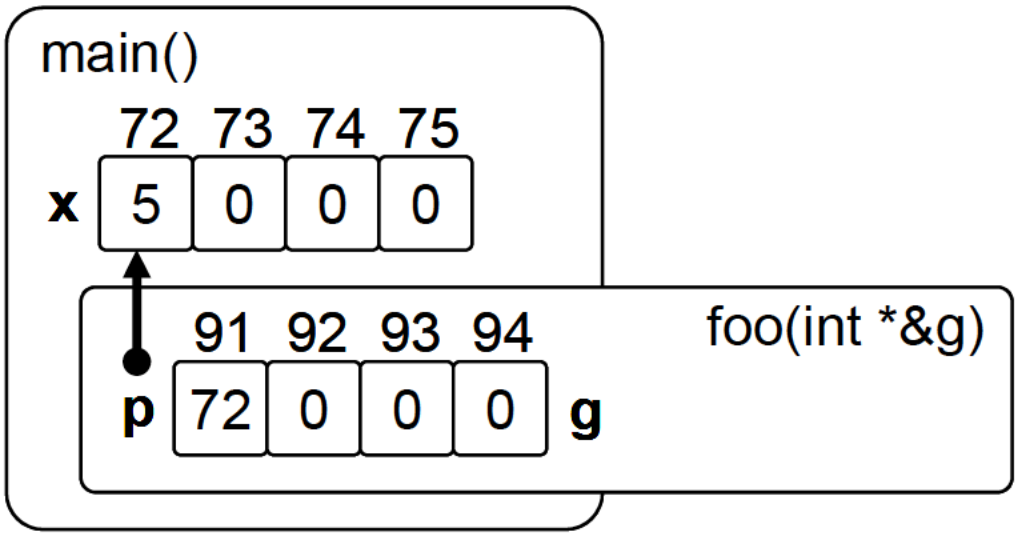


Pointers

```
void foo(int *&g)
{
        *g = *g + 1;
        g = g + 1
}

void main()
{
        int  x = 5;
        int  *p = &x;

        foo(p);
        foo(&x);


}
```

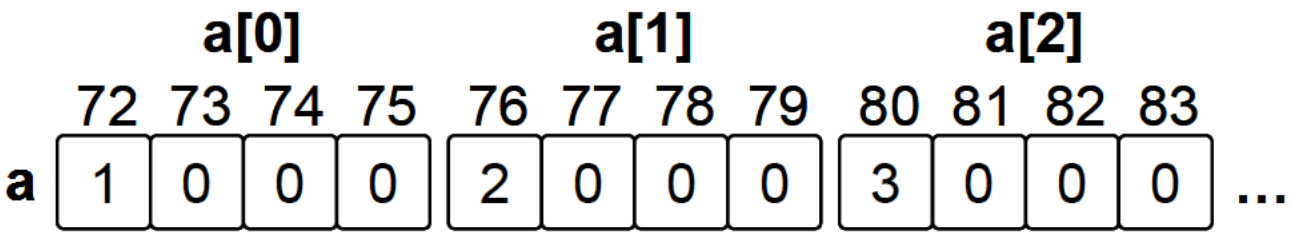
Pointers

```
void foo(int *&g)
{
        *g = *g + 1;
        g = g + 1
}

void main()
{
        int  x = 5;
        int  *p = &x;

        foo(p);
        foo(&x);      // Wrong!
        // value of x has changed
        // value of p has changed
}
```



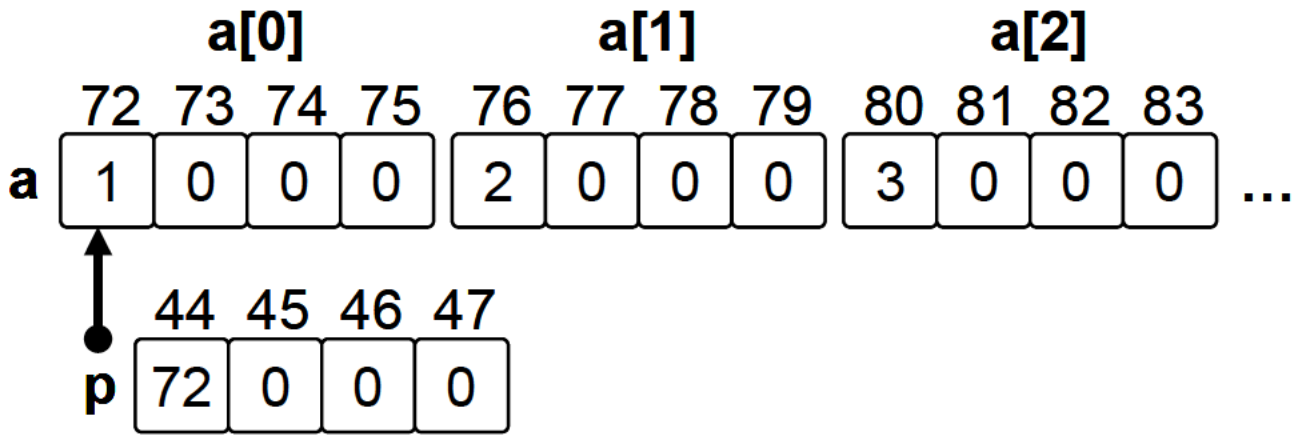
Pointers v.s Array

```
void main()
{
        int  a[10];
        printf("%d\n", a);
        printf("%d\n", &a[0]);      // a == &a[0].
}
```



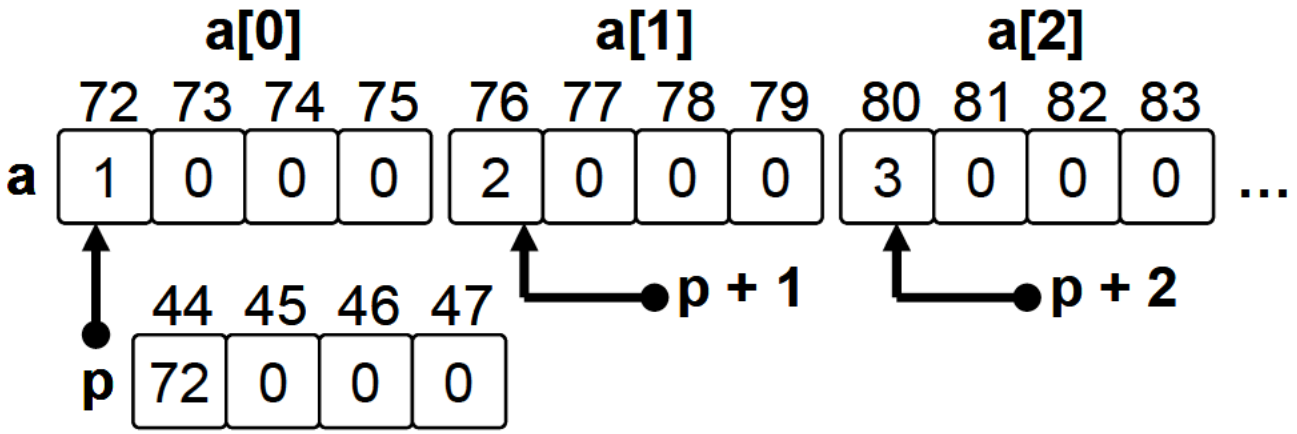
Pointers v.s Array

```
int  a[100] = { 1, 2, 3 };
int  *p = a;
*p = *p + 1;
printf("%d\n", *p);
```



Pointers v.s Array

```
int  a[100] = { 1, 2, 3 };
int  *p = a;
printf("%d\n", p + 1);
printf("%d\n", *(p + 2) );
```



Question 1

```
void main()
{
        int     *x, y;
        float  *z;

        y = 2;
        z = &y;
        *x = *x + y;
        printf("%d", y);
}
```

1. What is the problem in this program?
2. How to fix it?

Question 2

```
void main()
{
        double   m[100];
        double   *p1, *p2;

        p1 = m;
        p2 = &m[6];
}
```

1. How many bytes from p1 to p2?

Question 3

```
void main()
{
        int     x = 1023;
        char  *p = (char *)&x;

        printf("%d %d %d %d\n", p[0], p[1], p[2], p[3]);
}
```

1. What are the results in screen?

Question 4

```
void swap1(int x, int y)
{
        int temp = x;
        x = y;
        y = temp;
}
```

```
void swap2(int &x, int &y)
{
        int temp = x;
        x = y;
        y = temp;
}
```

```
void swap3(int *x, int *y)
{
        int temp = *x;
        *x = *y;
        *y = temp;
}
```

1. What is the difference among these three functions?

Questions

What is Data Structure?

Questions

What is Data Structure?

- A fundamental concept of any programming language, essential for algorithmic design.
- Used for the efficient organization and modification of data.
- How data and the relationship amongst different data is represented, that aids in how efficiently various functions or operations or algorithms can be applied.

Questions

How many types of Data Structures?

Questions

How many types of Data Structures?

Two types of data structures:

Questions

How many types of Data Structures?

Two types of data structures:

1. **Linear data structure:** If the elements of a data structure result in a sequence or a linear list then it is called a linear data structure. Example: arrays, linked list, stacks, queues.

Questions

How many types of Data Structures?

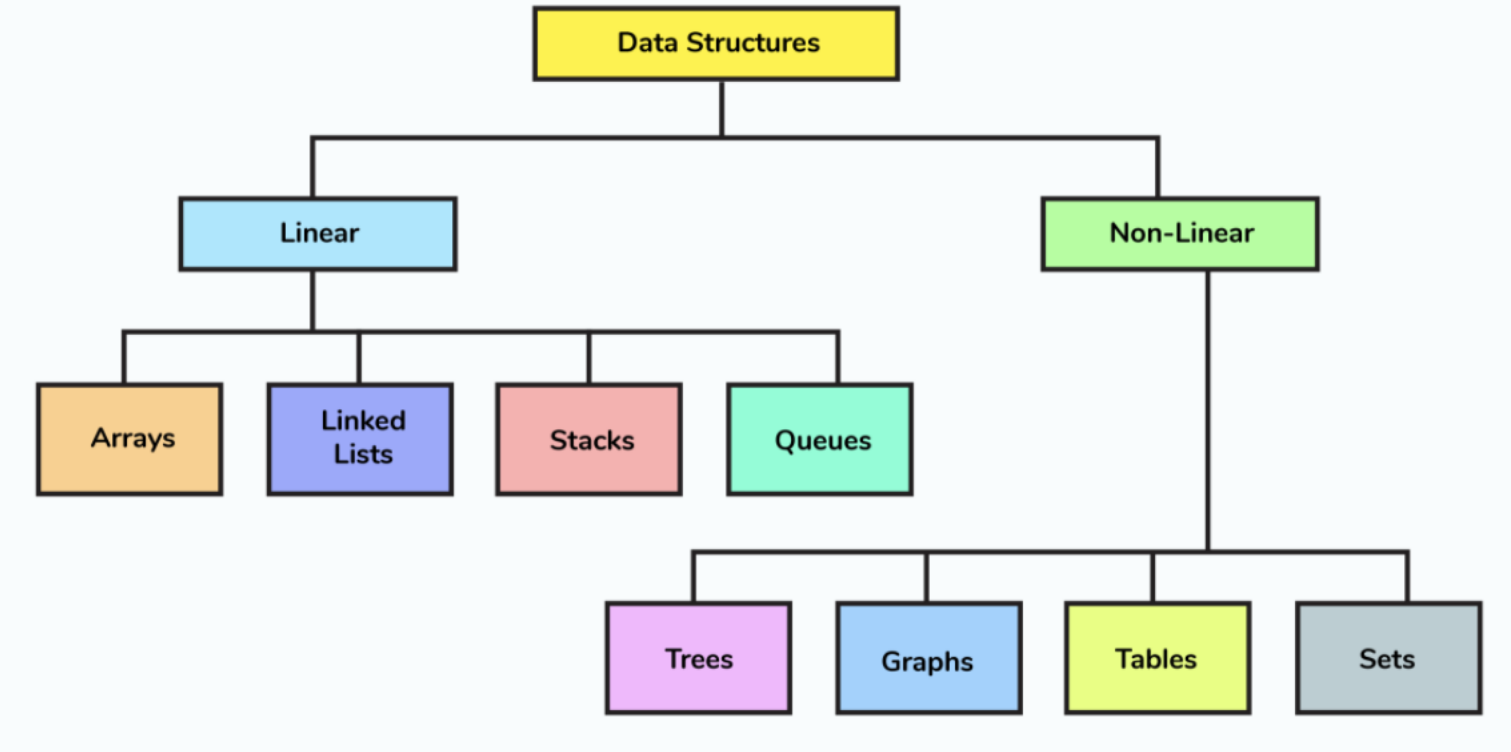
Two types of data structures:

1. **Linear data structure:** If the elements of a data structure result in a sequence or a linear list then it is called a linear data structure. Example: arrays, linked list, stacks, queues.
2. **Non-linear data structure:** If the elements of data structure results in a way that traversal of nodes is not done in a sequential manner, then it is a non-linear data structure. Example: trees, graphs, etc.

Questions

How many types of Data Structures?

Two types of data structures:



Source: https://www.interviewbit.com/data-structure-interview-questions/

Questions

Applications of Data Structures?

Questions

Applications of Data Structures?

- Artificial intelligence
- Compiler design
- Machine learning
- Database design and management
- Blockchain Numerical and Statistical analysis
- Operating system development
- Image & Speech Processing
- Cryptography

Questions

What is an array?

Questions

What is an array?

- The collection of similar types of data stored at contiguous memory locations.
- The simplest data structure where the data element can be accessed randomly just by using its index number.

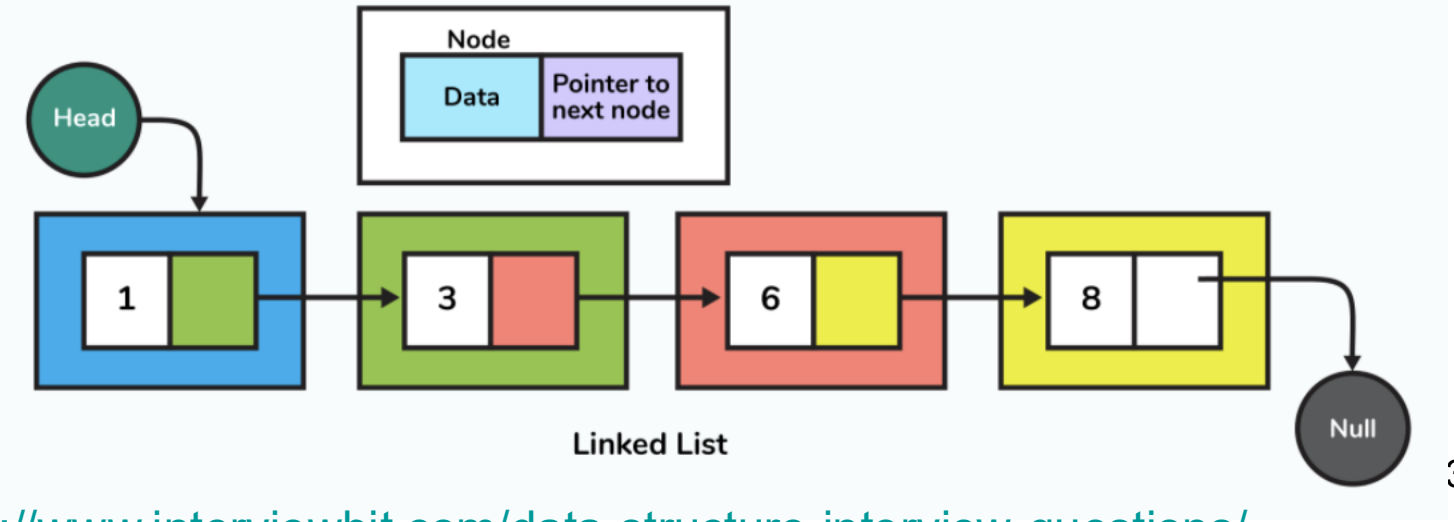
Questions

What is a linked list?

Questions

What is a linked list?

- A data structure that has sequence of nodes where every node is connected to the next node by means of a reference pointer.
- The elements are not stored in adjacent memory locations. They are linked using pointers to form a chain.
- This forms a chain-like link for data storage.



Source: https://www.interviewbit.com/data-structure-interview-questions/

Questions

Are linked lists of linear or non-linear type?

Questions

Are linked lists of linear or non-linear type?

- They can be considered both linear and non-linear data structures. This depends upon the application that they are used for.

Questions

Are linked lists of linear or non-linear type?

- They can be considered both linear and non-linear data structures. This depends upon the application that they are used for.
- When linked list is used for access strategies, it is considered as a linear data-structure.

Questions

Are linked lists of linear or non-linear type?

- They can be considered both linear and non-linear data structures. This depends upon the application that they are used for.
- When linked list is used for access strategies, it is considered as a linear data-structure.
- When they are used for data storage, it can be considered as a non-linear data structure.

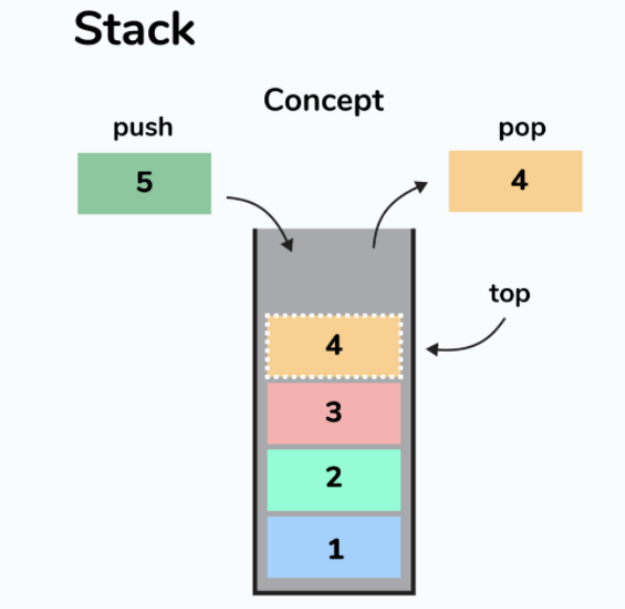
Questions

What is a stack? What are the applications of stack?

Questions

What is a stack? What are the applications of stack?

- Stack is a linear data structure that follows LIFO (Last In First Out) approach for accessing elements.
- Push, pop, and top (or peek) are the basic operations of a stack.



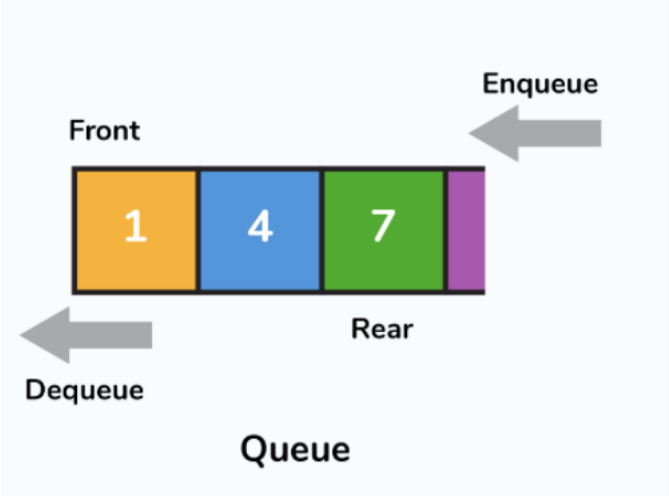
Questions

What is a queue? What are the applications of queue?

Questions

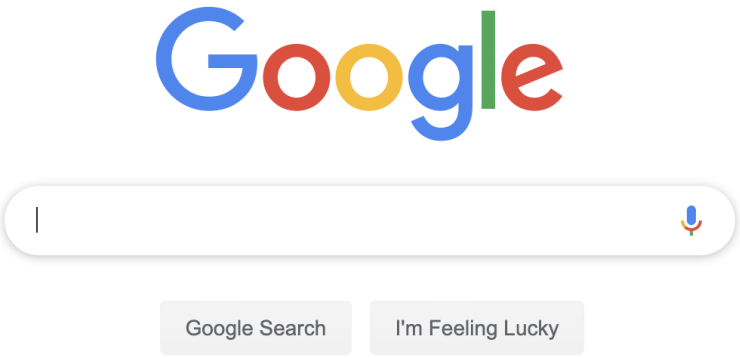
What is a queue? What are the applications of queue?

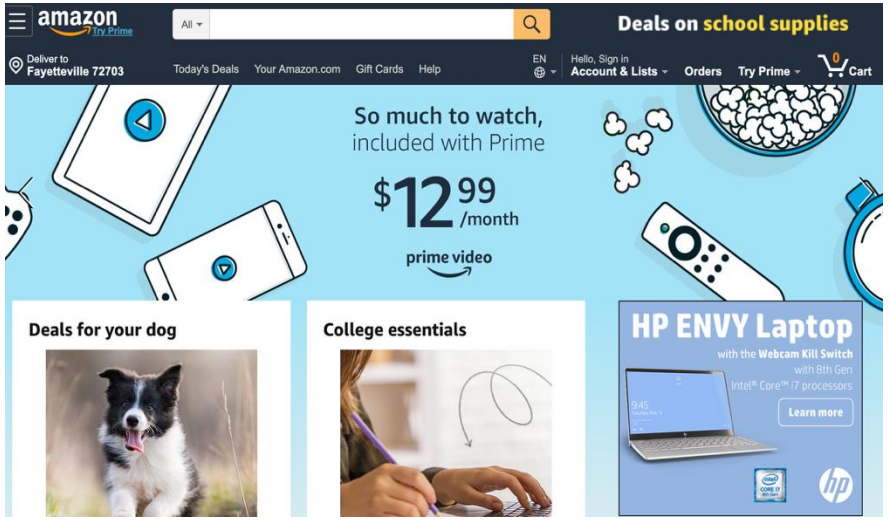
- A queue is a linear data structure that follows the FIFO (First In First Out) approach for accessing elements.
- Dequeue from the queue, enqueue element to the queue, get front element of queue, and get rear element of queue are basic operations that can be performed.



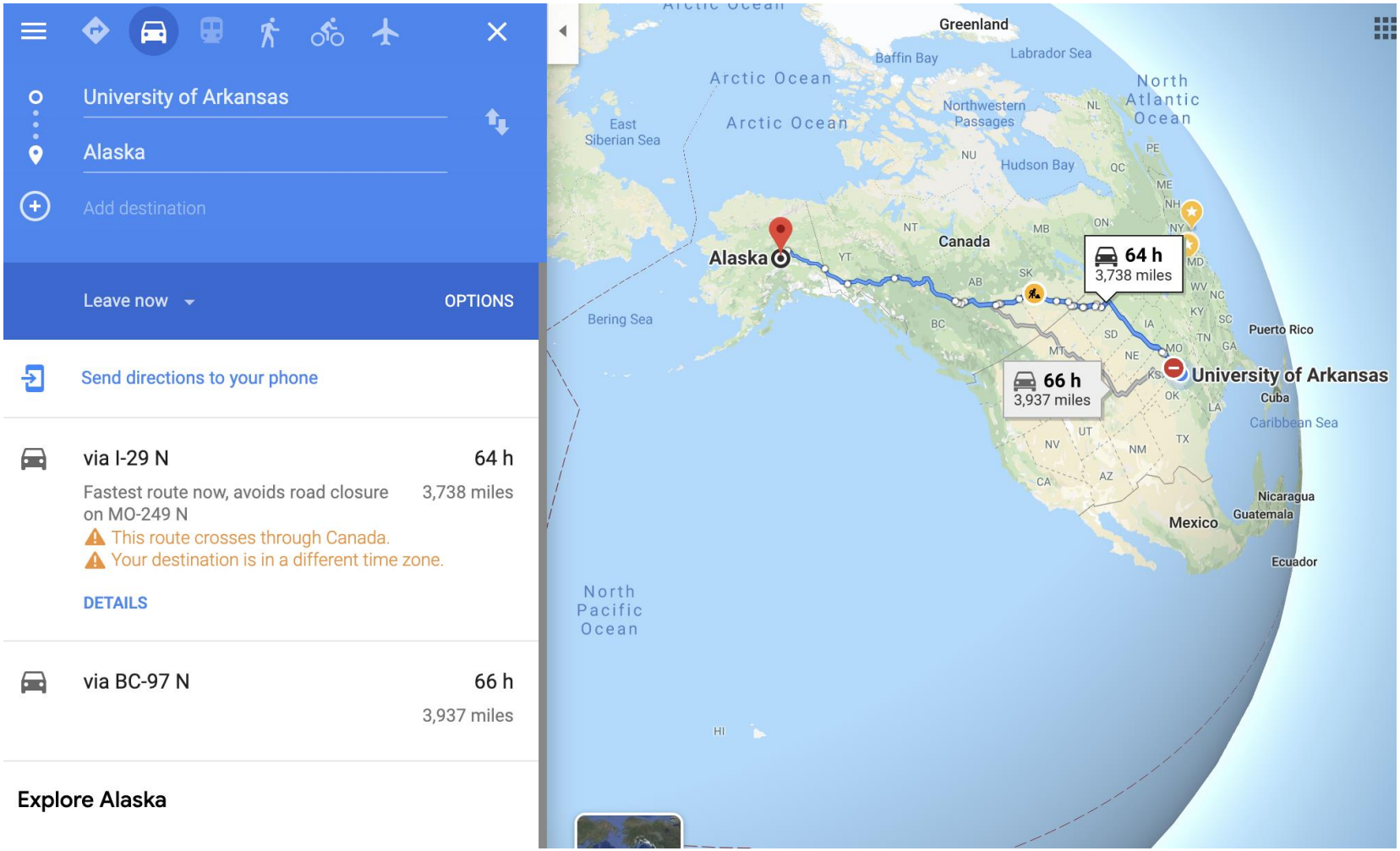
Why Data Structure is Important?

Data Structure is the Key

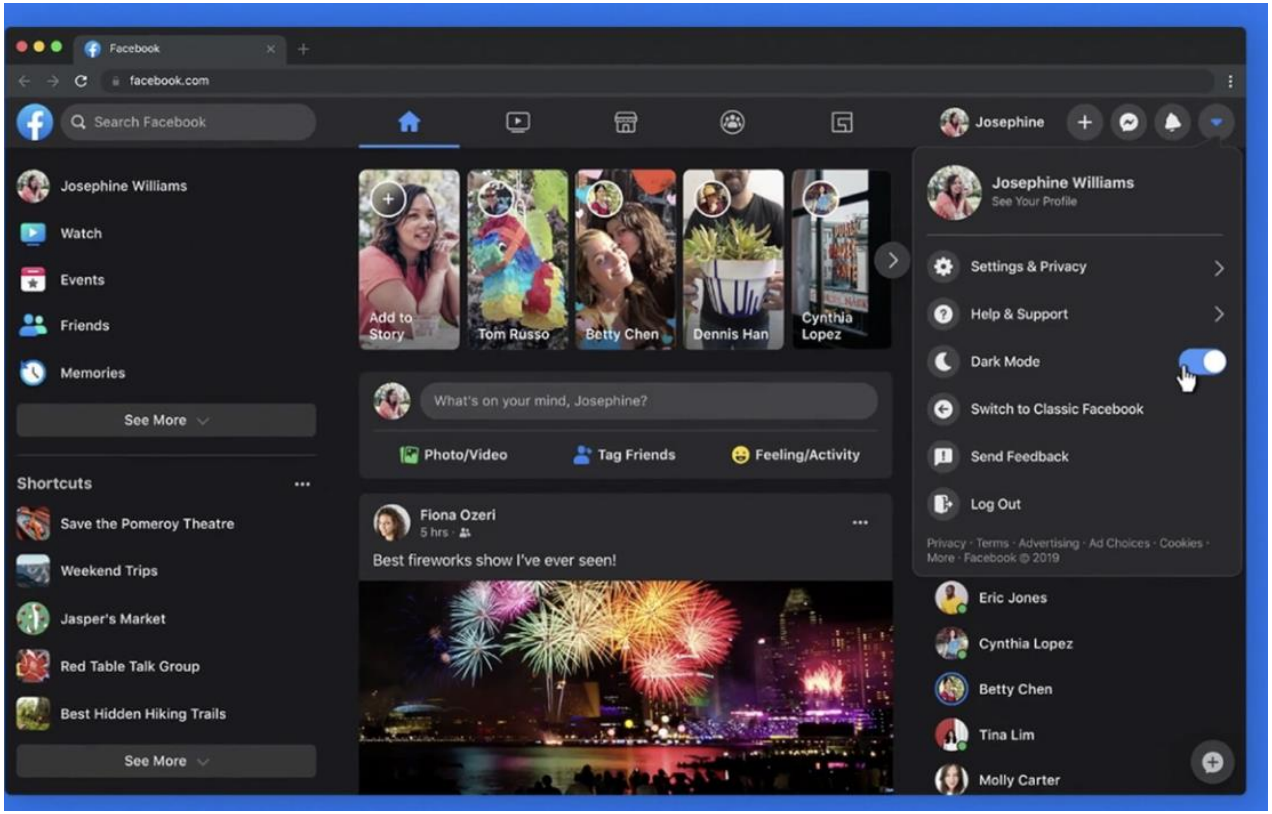




Data Structure is the Key



Data Structure is the Key



Multiple Object Tracking in Graph