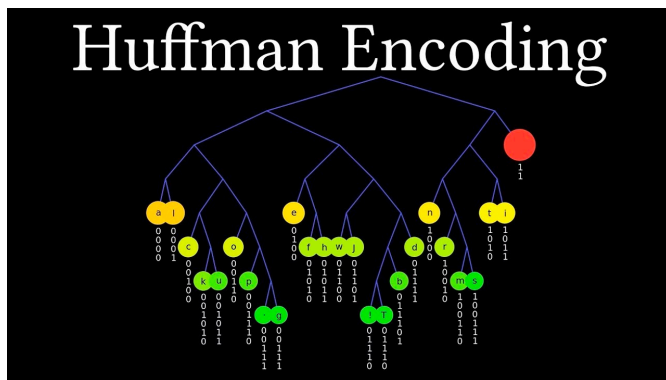
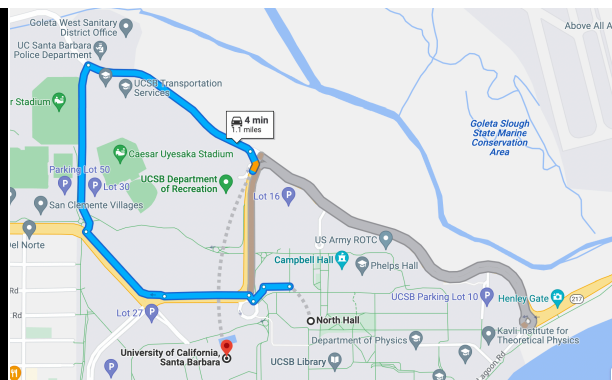


# PRACTICE WITH PRIORITY QUEUES & HASHTAEBLE

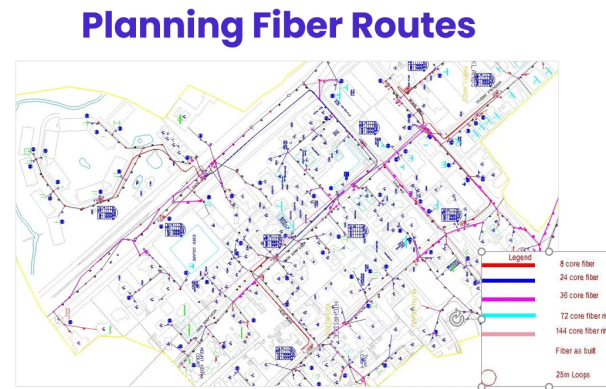
Problem Solving with Computers-II



**Data Compression**



**Navigation**



**Network Design**

# Web traffic analysis

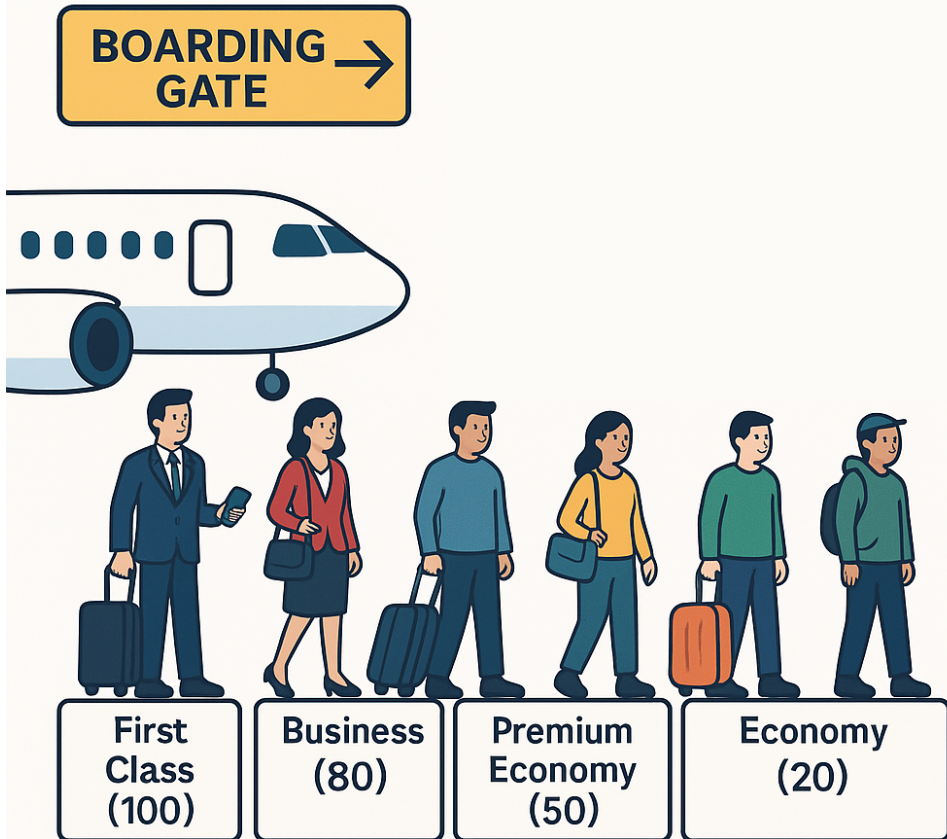
You're back at your job at Amazon. Your boss hands you a log of IP addresses from today's traffic and asks:

1. How many unique visitors did we have?
2. How many times did each IP visit?
3. Which IP visited the most?
4. **Who are our top K most frequent visitors?**
5. Flag any suspicious IPs (visited > threshold times)

Open [traffic\\_activity.cpp](#)

See code from lecture where we explored multiple approaches to Q4.

# C++ Priority Queue ADT



```
priority_queue<int> pq;
```

```
pq.push(20);  
pq.push(20);  
pq.push(80);  
pq.push(50);  
pq.push(100);
```

```
cout << pq.top();
```

```
pq.pop();
```

# Configuring std::priority\_queue

```
template <
    class T,
    class Container= vector<T>,
    class Compare = less <T>
> class priority_queue;
```

The template for priority\_queue takes 3 arguments:

1. Type elements contained in the queue.
2. Container class used as the internal store for the priority\_queue, the default is **vector<T>**
3. Class that provides priority comparisons, the default is **less**

# Configuring std::priority\_queue

**//Configure for a max-heap**

```
priority_queue<int, vector<int>, std::less<int>> pq;
```

**//Configure for a min-heap**

```
priority_queue<int, vector<int>, std::greater<int>> pq;
```

## Trace the output of this code

```
int arr[]={10, 2, 80};
priority_queue<int*> pq;
for(int i=0; i < 3; i++)
    pq.push(arr+i);

while(!pq.empty()){
    cout<<*pq.top()<<endl;
    pq.pop();
}
```

How can we change the way pq  
prioritizes pointers?

## Write a comparison class to get the desired output

```
class cmpPtr{
    bool operator()(int* a, int* b) const {
        return _____;
    }
};

int arr[]={10, 2, 80};
priority_queue<int*, vector<int*>, cmpPtr> > pq;
for(int i=0; i < 3; i++)
    pq.push(arr+i);

while(!pq.empty()){
    cout<<*pq.top()<<endl;
    pq.pop();
}
```

```
Output: 80
        10
        2
```

# 215. Kth Largest Element in an Array

Medium

📁 Topics

🏢 Companies

Given an integer array `nums` and an integer `k`, return the  $k^{\text{th}}$  largest element in the array.

Note that it is the  $k^{\text{th}}$  largest element in the sorted order, not the  $k^{\text{th}}$  distinct element.

Can you solve it without sorting?

## Example 1:

Input: `nums = [3,2,1,5,6,4]`, `k = 2`

Output: 5

## Example 2:

Input: `nums = [3,2,3,1,2,4,5,5,6]`, `k = 4`

Output: 4

Activity (10 mins): Brainstorm and describe a possible solution in plain English

Trace your approach on the example inputs

Kth Largest Element in an Array (medium):

<https://leetcode.com/problems/kth-largest-element-in-an-array/description/>

# Leetcode practice (LP04)

LP04 (PQ + Hashtables): <https://ucsb-cs24.github.io/s25/lp/lp04/>

## Priority Queues must know problems:

1. Kth Largest Element in an Array (medium):  
<https://leetcode.com/problems/kth-largest-element-in-an-array/description/>
  2. Top K Frequent Elements (medium):  
<https://leetcode.com/problems/top-k-frequent-elements/description/>
- \* Practice configuring a PQ in different ways using a comparison class