Name: Esteban Escartin
Email: eescartin@csu.fullerton.edu
Assignment: Project 2

Pseudocode

Top k most frequent elements:
Time: O(n log k)
Space: O(n + k)

```
Def top k(nums, k):
    make a dict to store frequency
    for x in nums:
        add 1 to x in dictionary
    make heap
    for num in dict.keys():
        freq = dict[nums]
        heappush(freq, num)
        if len(heap) > k:
            heap pop min element
    answer = []
    while heap:
        answer.add( pop min from heap )
    answer.reverse()
    return answer
```

Network delay time:
Time: O(E + V log V)
Space: O(E + V)

```
Def delay(times, n, k):
    adjacency list (dict)
    for u, v, w in times:
```

```
        graph[u] = []
        graph[u].append((v,w))
dists = [INT MAX for _ in range(n+1)]
dists[k] = 0
q = queue([(0,k)])
while q:
        time , curr = q pop left
        if curr in graph:
                for neighbor, delay in graph[curr]:
                        if time + delay < dists[neighbor]:
                                dists[neighbor] = time + delay
                                q.append((time+delay, neighbor))
Max_delay = max_element(dists[1:])
Return max_delay if max_delay != INT MAX else -1
```