LaTeX Notes CIS 5210

Jonathon Delemos

May 18, 2025

This is a note repositrory.

1 Section One: Python

1.1 Google Colab

`We will be using Google Colab for all course preparations.`

2 Section Two: Algorithms and Data Structures

2.1 Algorithm 1: Binary Search

```
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

2.2 Algorithm 2: Merge Sort

```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]

        merge_sort(L)
        merge_sort(R)
```

```
        i = j = k = 0

        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1

        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1

        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1
arr = [38, 27, 43, 3, 9, 82, 10]
merge_sort(arr)
print("Sorted array is:", arr)
```

2.3 Algorithm 3: BFS

```
from collections import deque
def bfs(graph, start):
    visited = set()
    queue = deque([start])
    visited.add(start)

    while queue:
        vertex = queue.popleft()
        print(vertex, end=" ")

        for neighbor in graph[vertex]:
            if neighbor not in visited:
                visited.add(neighbor)
                queue.append(neighbor)
graph = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
    'D': ['B'],
```

```
    'E': ['B', 'F'],
    'F': ['C', 'E']
}
bfs(graph, 'A')
```