3 Asymptotic Complexity, Running Time Analysis, Graphs

- Math background
  - Asymptotic Notation
  - Running time analysis
  - Graphs
- Graph algorithms

3.1 Asymptotic Complexity

Definition 3.1
We say $f(n) \in O(g(n))$ if $\exists c > 0, n_0 \geq 0$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$

We say $f(n) \in \Omega(g(n))$ if $\exists c > 0, n_0 \geq 0$ such that for all $n \geq n_0, f(n) \geq c \cdot g(n)$

Equivalently, $g(n) \in O(f(n))$.

We say $f(n) \in \Theta(g(n))$ if $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$

Example 3.2
For the expression $10n^2$, we can see that:

$$\begin{aligned} 10n^2 &\in O(n^2), O(n^3), \ldots \\ &\in \Omega(n^2), \Omega(n) \end{aligned}$$

We can also express strict bounds using o and ω notation:

Definition 3.3
We say $f(n) \in o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

We say $f(n) \in \omega(g(n))$ if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

Example 3.4
We can see that $10n^2 \in o(n^3)$, because

$$\lim_{n\to\infty} \frac{10n^2}{n^3} = \lim_{n\to\infty} \frac{10}{n} = 0$$

3.2 Running time analysis

We need to count the elementary steps of an algorithm, which can depend on the data structures used. Some common running times:

- Logarithmic time, $O(\log n)$. Example: binary search
- Linear time, $O(n)$. Example: max/min of a list, etc.
- Polynomial time $O(n^k)$ for some constant k.

  Example: search over subsets of size k of a set size n, where the number of subsets is:

  $$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots(2)(1)}$$

  Which leads to a running time of around $\Theta(n^k)$.

- Exponential time, e.g., $O(2^n)$. Note that something like 3^n grows asymptotically faster than 2^n, so their time complexities would not be equivalent.

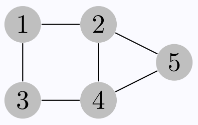
  Example: searching over all subsets of a set of size n.

3.3 Graphs

Definition 3.5
An **undirected graph** $G = (V, E)$ is a finite set V of **vertices** and E of **edges**, which are unordered pairs of vertices.

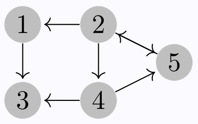
A **directed graph** (or digraph) $G = (V, E)$ is a finite set V of **vertices** and a set E of **edges**, which are ordered pairs of vertices.

Example 3.6



In this graph, we have

$$V = \{1, 2, 3, 4, 5\}$$
$$E = \{\{1, 2\}, \{1, 3\}, \cdots\}$$



In this graph, we have

$$V = \{1, 2, 3, 4, 5\}$$
$$E = \{(2, 1), (1, 3), (2, 5), (5, 2), \cdots\}$$

Note 3.7
Notice that

- There are no multiple edges
- Digraphs can have self loops, because edges are denoted as ordered pairs. But the way we have defined our undirected graphs, we can not have self loops because the edges are all sets of 2 numbers.

3.3.1 Graph Terminology

- Vertices are sometimes called **nodes**
- Edge $\{u, v\}$ **joins** vertices u and v, which are its **ends**; u and v are **adjacent**; the edge is **incident** on the vertices
- Edge (u, v) has **tail** u and **head** v
- The **degree** of a vertex, $\deg(v)$, is its number of incident edges
- In a digraph, $\text{indeg}(v)$ (**indegree** of v) is the number of edges with v as the head, and $\text{outdeg}(v)$ (**outdegree** of v) is the number of edges with v as the tail

- $n = |V|$, $m = |E|$

$$\sum_{v \in V} \deg(v) = 2m$$
$$\sum_{v \in V} \text{indeg}(v) = \sum_{v \in V} \text{outdeg}(v) = m$$

- In an undirected graph, $m \leq \binom{n}{2} = O(n^2)$. In a digraph, $m \leq n^2$
- A subgraph of $G = (V, E)$ is a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$. Must make sure that there are no edges in E' that do not connected nodes that are not in V'.

3.3.2 Graph Representations

- Adjacency matrix: $|V| \times |V|$ matrix indexed by vertices, with

$$A_{uv} = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases}$$

- Adjacency list: for each $v \in V$, give a list Adj_v of all its (outgoing) neighbors. This is a more efficient representation for graphs that do not have many edges.
- Note that the adjacency matrix would have $\Theta(n^2)$ of size, and the adjacency list would have $\Theta(n + m)$ size.

3.3.3 Paths and Cycles

Definition 3.8
A **path** is a sequence of vertices $(v_0, v_1, \cdots, v_k)$ such that $(v_i, v_i + 1) \in E$ for all $i \in \{0, 1, \cdots, k - 1\}$

The **length** of a path is its number of edges, k.

A path is **simple** if all its vertices and edges are distinct.