Recall let $\text{func}$ be an alg, and $I_n$ the set of its inputs of size $n$.

$$WC_{\text{func}}(n) = \max \left\{ \text{runtime of } \text{func}(x) \mid x \in I_n \right\}$$

worst-case

Sometimes worst-case is too pessimistic.

We define the average-case runtime as

$$AVG_{\text{func}}(n) = \frac{\sum \text{runtime of } \text{func}(x) \mid x \in I_n}{|I_n|}$$
Example

def search_one(lst: List[int]) -> bool:
    for num in lst:
        if num == 1:
            return True
    return False

For this example, we'll consider only the inputs

\( I_n = \text{the permutations of the numbers} \{1, 2, \ldots, n\} \)

\( I_0 = \{[1]\} \)

\( I_1 = \{[2]\} \)

\( I_2 = \{[1,2], [2,1]\} \)

\( I_3 = \{[1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]\} \)
Fact: \( \forall n \in \mathbb{N}, \ |I_n| = n! = \frac{n!}{\prod_{i=1}^{n} i} \)

Analysis (average-case)

From the definition:

\[
\text{Avg}_{s_0}(n) = \sum_{\text{lst } \in I_n} \frac{\text{runtime of search\_one}(\text{lst})}{n!} \]

For any lst with \( \text{lst}[i] = 1 \), there are \( i+1 \) iterations of the loop, taking \( i+1 \) steps total.

Rough:

- \( 5 \ 1 \ 2 \ 3 \ 4 \) \( \rightarrow \) 2 iters
- \( 1 \ 2 \ 3 \ 5 \ 4 \) \( \rightarrow \) 1 iter
- \( 4 \ 3 \ 2 \ 5 \ 1 \) \( \rightarrow \) 5 iters

\( 0 \ 1 \ 2 \ \ldots \ \frac{1}{i} \ \ldots \ \frac{1}{n-1} \)
Key idea: group inputs in $I_n$ based on where the 1 is.

$$\text{Avg}_{0:n}(n) = \frac{\sum \text{ r.t. of search-one}(1st)}{n!}$$

$$= \frac{1}{n!} \sum_{i=0}^{n-1} \left( \sum_{1 \leq j \leq n} \text{ r.t. of search-one}(1st) \right)_{1 \leq i \leq n, \overline{i}\{i\} = 1}$$

$$= \frac{1}{n!} \sum_{i=0}^{n-1} \left( \sum_{1 \leq j \leq n} (i+1) \right)_{1 \leq i \leq n, \overline{i}\{i\} = 1}$$

$2, 3, \ldots, n^2$

Key idea #2: for each $i$, there are $(n-1)!$ input lists where 1 is at index $i$.

Continuing from above:

$$\text{Avg}_{0:n}(n) = \frac{1}{n!} \sum_{i=0}^{n-1} (i+1)(n-1)!$$
\[
\frac{(n-1)!}{n!} \sum_{i=0}^{n-1} (i+1)
\]

\[\text{course notes}\]

\[= \frac{n+1}{2} \in \Theta(n)\]

---

Graphs! (Chapter 6)

Recall:

\[G = (V, E)\]

\[\text{graph} \quad \text{vertices} \quad \text{edges}\]

Theorem:

\[\forall G = (V, E), \quad |E| \leq \frac{|V|(|V|-1)}{2}\]

Proof:

Let \(G = (V, E)\) be an arbitrary graph.

We'll prove that \(|E| \leq \frac{|V|(|V|-1)}{2}\).
Key idea: an edge is a set of 2 vertices i.e., a subset of $V$ of size 2.

So,

$$|E| \leq \# \text{ of subsets of size 2 of } V$$

$$= \frac{|V|(|V|-1)}{2}$$

(see past worksheet)

---

**Connectedness**

Let $G = (V, E)$. Let $u, v \in V$.

We say $u$ and $v$ are connected in $G$ if and only if there exists a path between them.
We say $G$ itself is connected if and only if all pairs of vertices are connected.

$G$ is connected: $\forall u,v \in V, \text{Conn}(G,u,v)$

"$u$ and $v$ are connected in $G$"

Interesting Question:

Given a graph $G$, determine whether it is connected.

- develop an algorithm (CSC 263/373)
- determine a condition that implies that a graph is connected.

\[
\frac{(1+1-1)(1+1-2)}{9}
\]
**Theorem**

\[ \forall G = (V, E), \quad |E| \geq \frac{(|V|-1)(|V|-2)}{2} + 1 \implies \]

**G is connected.**

**Proof**

**Base case** (see course notes)

**Induction step** Let \( k \in \mathbb{Z}^+ \). Assume \( P(k) \), and we'll prove \( P(k+1) \).
Let \( G = (V, E) \). Assume \( |V| = k+1 \).
And assume \( |E| \geq \frac{k(k-1)}{2} + 1 \).
We'll prove \( G \) is connected.

(Rough)
Pick a vertex \( v \).
Remove \( v \) from \( G \) to obtain a new graph \( G' = (V', E') \).

We know
\[
|V'| = |V| - 1 = k
\]
\[
|E'| = |E| - \# \text{ edges on } v
\]

To use I.H., first need to check that
\[
|E'| \geq \frac{(k-1)(k-2)}{2} + 1
\]

trickier than it seems!

If we can do this, we conclude that
$G'$ is connected. (by I.H.)

Since $v$ is connected to $G'$, the entire graph $G$ is connected. To prove this!