Branch-\&-Bound

*algorithms* for discrete optimization problems:

**P**: find $Z^* = \text{Minimum} \{cx : x \in S\}$

where the feasible set $S$ is discrete,

for example, $S = \{x : Ax \geq b, x \in \{0,1\}\}$
In order to visualize the branch-and-bound approach, we will use a **search tree**:

- Each **node** of the search tree for a problem represents a *subset* of feasible solutions of the problem.
- The **root** of the tree represents the set of all feasible solutions of the problem.
- The **descendants** of each node of the tree represent a *partition* of the set represented by that node.
Creating descendents of a node with feasible set \( S_t \) is done by partitioning.

A collection of subsets \( S_i, i = 1, 2, \ldots, t \) of set \( S \) is a **partition** of \( S \) if

\[
S_1 \cup S_2 \cup \cdots \cup S_t = S
\]

and

\[
S_i \cap S_j = \emptyset, \text{ i.e., they are mutually disjoint}
\]
A **leaf node** is a node without descendents. A **terminal node** is a leaf node whose feasible set consists of a single solution, i.e., a singleton.

A node can be **fathomed** (**pruned**) if one of three conditions are met:

- **Pruning by optimality:** $z_t = \min \{ cx : x \in S_t \}$ has been computed.
- **Pruning by bound:** the inequality $z \leq \min \{ cx : x \in S_t \}$ has been proved.
- **Pruning by infeasibility:** $S_t = \emptyset$

in which case it is a leaf node, i.e., it is not necessary to create descendents (**branch**).
If no pruning is done, then all the terminal nodes must be determined by \textit{complete enumeration}.

By pruning, we perform \textit{implicit enumeration}, i.e., we avoid explicitly finding each terminal node.

Pruning by bound, i.e., proving that $z \leq \min\{cx : x \in S_t\}$, is usually achieved by using a \textit{relaxation} of $\min\{cx : x \in S_t\}$.
Consider a constrained optimization problem

\[ P: \quad z^* = \min \{ cx \mid x \in S \} \]

and a problem

\[ P^R: \quad z^R = \min \{ f(x) \mid x \in T \} \]

The problem \( P^R \) is a relaxation of problem \( P \) if:

- \( S \subseteq T \), i.e., every \( x \) feasible in \( P \) is also feasible in \( P^R \), and

- \( f(x) \leq cx \quad \forall x \in S \)
**Proposition:** If $P^R$ is a relaxation of $P$, then its optimal value is a lower bound of the optimal value of $P$:

\[ z^R \leq z^*. \]

Therefore, if at some node we solve a relaxation $P^R$ and find $z^R$ and can show that

\[ z^R \geq z^*, \]

we can **prune** the node.

To be useful, $P^R$ should be easier to solve than $P$. 
Linear Programming Relaxation of Integer (& Mixed-Integer) LP

The most common relaxation of IP & MIP problems is the LP relaxation, in which integer restrictions are removed. Suppose that

\[ P: \quad z = \min \{ cx \mid Ax \geq b, x \in Z_+^n \} \]

where \( Z_+^n \) is the set of \( n \)-dimensional vectors of non-negative integers. The LP relaxation is

\[ P^{LP}: \quad z^{LP} = \min \{ cx \mid Ax \geq b, x \in R_+^n \} \]

where \( R_+^n \) is the set of \( n \)-dimensional real non-negative vectors.
**Note:** in the above definition of *relaxation*, let

\[ f(x) = cx \text{ and} \]

\[ S \equiv \left\{ x \mid Ax \geq b, x \in \mathbb{Z}^n_+ \right\} \quad \& \quad T \equiv \left\{ x \mid Ax \geq b, x \in \mathbb{R}^n_+ \right\} \]

so that \( S \subset T \)

That is, while the objective functions of \( P \) & \( P^{LP} \) are the same, relaxing the integer restrictions of an IP adds feasible solutions to the problem, so that a lower minimum might be found.