Tries to represent strings (digital trees) \( \Sigma \): fixed alphabet 

cave, bat, at

A tree to store the set of strings and each node has arity \( |\Sigma| \)

Shing matching,
Common substrings
Longest common prefix

How efficiently can you construct Tries

In particular \( O(|\Sigma| \leq k_i) = O(N) \)

For \( |\Sigma| \leq N \), we could sort in optimal time.

How about space? \( |\Sigma| N \)
Knapsack problem

Given a knapsack of capacity $B$ and $n$ items with profits $p_i$, $i \leq n$ and weights $w_i$, $i \leq n$ we want to fill up the knapsack so that we maximize profit of the items filled. Let

\[ x_i = \begin{cases} 
1 & \text{if item } i \text{ is included} \\
0 & \text{otherwise}
\end{cases} \]

Maximize \[ \sum x_i p_i \]

s.t. \[ \sum x_i w_i \leq B \]
\[ x_i \in \{0, 1\} \]

E.g. \[ B = 15 \quad n = 4 \]
\[ p_i = 10 \quad 10 \quad 12 \quad 18 \]
\[ w_i = 2 \quad 4 \quad 6 \quad 9 \]
\[ R_i = 5 \quad 2.5 \quad 2 \quad 2 \]
Greedy approach

(Residual capacity, item to be considered, current profit)

\[(15, i_1, i_2, i_3, 0) \rightarrow (6, i_1, i_2, 33, 18)\]

\[\rightarrow (0, i_1, 2, 30)\]

Straightforward greedy doesn't get us the best solution for some instances

Even very greedy with ratio will not necessarily work

If we are willing to reconsider past decisions?
- With backtracking, we can obtain the exact solution; however, we may visit all leaf nodes \( \Rightarrow \) brute force \( 2^n \) algo.

- Can we prune the search?

  can we avoid visiting all nodes and still be convinced about the final result?

**Can we "estimate" the profit possible by visiting a subtree?**

In particular, an upper bound \( U(v) \).

At any node \( v \) of the search tree, we have a lower bound \( L(v) \) and an upper bound \( U(v) \).

- The best that we can do by visiting \( v \) is no worse than exploring its subtree in \( v \).

\[ L(v) > U(v) \]
Even if we were to use greedy strategy, there is no provable guarantee that we will not visit all the $2^n$ nodes.

A * algorithm

Greedy works for Minimal Spanning Trees