Exercises

1. Suppose that $S = \{1, 2, \cdots, n\}$. Write a backtracking algorithm to generate all permutations of S.
2. Suppose that $S = \{1, 2, \cdots, n\}$. Write a backtracking algorithm to generate all k-subsets of S.
3. (Set Cover Problem) Suppose that $S = \{1, 2, \cdots, n\}$ and $C \subseteq Powerset(S)$ is a collection of subsets of S. Write a backtracking algorithm to find a $C' \subseteq C$ such that:

$$\bigcup_{c \in C'} c = S$$

, where $|C'|$ is minimum.

4. Devise a backtracking algorithm to solve the SUDOKU puzzle from an initial state.
5. A derangement is a permutation p of $\{1, 2, \cdots, n\}$ such that no item is in its proper position, i.e. $p_i \neq i$ for all $1 \leq i \leq n$. Write a backtracking program that constructs all the derangements of n items.
6. For a given number n, write a backtracking algorithm to generate all it partitions, i.e.

$$\begin{aligned} 4 &= 1+1+1+1, \\ 4 &= 2+1+1, \\ 4 &= 2+2, \\ 4 &= 3+1, \\ 4 &= 4. \end{aligned}$$

Exercises

7. (Set Cover Problem) Suppose that $S = \{1, 2, \cdots, n\}$ and $C \subset Powerset(S)$ is a collection of subsets of S. Write a Branch and Bound algorithm to find a $C' \subset C$ such that:

$$\bigcup_{c \in C'} c = S$$

, where $|C'|$ is minimum.

8. Suppose that $G = (V, E)$ is a graph. A Clique is a complete subgraph of G. A Max-Clique is a clique containing maximum vertices.
   a. Write a Backtracking algorithm to find a Max-Clique of G.
   b. Change the Backtracking algorithm to Branch and Bound algorithm to find a Max-Clique of G.

Exercises

9. Describe the Adjacency Multi-list representation of a graph.
10. The square of a directed graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$ such that $(u, w) \in E^2$ if and only if for some $v \in V$, both $(u, v) \in E$ and $(v, w) \in E$. That is, G^2 contains an edge between u and w whenever G contains a path with exactly two edges between u and w. Describe efficient algorithms for computing G^2 from G for both the adjacency-list and adjacency-matrix representations of G. Analyze the running times of your algorithms.
11. The incidence matrix of a directed graph $G = (V, E)$ is a $|V| \times |E|$ matrix $B = (b_{ij})$ such that

$$b_{ij} = \begin{cases} -1 & \text{if edge } j \text{ leaves vertex } i, \\ 1 & \text{if edge } j \text{ enters vertex } i, \\ 0 & \text{otherwise.} \end{cases}$$

    Describe what the entries of the matrix product $B \times B^T$ represent, where B^T is the transpose of B.
12. The diameter of a tree $T = (V, E)$ is given by

$$\max \delta(u, v) \; ; \; u, v \in V$$

    that is, the diameter is the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyze the running time of your algorithm.