From Book
Fundamentals of computer algorithms
pg. 368, 369

Algorithm RBackTrack(k)

RB(1)

n = h

```
{
  for (each x[k] ∈ T(x[1], ..., x[k-1]) do
  {
    if (B_k(x[1], x[2], ..., x[k]) ≠ 0) then
    {
      if (x[1], x[2], ..., x[k] is a path to an answer node)
      then      write (x[1:k]);
      if (k < n) then RBackTrack(k+1); //
    }
  }
}
```

General Recursive algorithm for BackTracking

Algorithm IBacktrack(n)

```
{
    k = 1;
    while (k ≠ 0) do
    {
        if (there remains an untried x[k] ∈ T(x[1], x[2], ..., x[k-1]) and
                Bk(x[1], ..., x[k]) is true) then
        {
            if (x[1], ..., x[k] is a path to an answer state)
                then write (x[1:k]);
            k = k+1;
        }
        else
        {
            k = k-1; // Backtracking to prev
        }
    }
}
```

0/1 knapsack problem solⁿ using Backtracking

Refer state space tree diagram from folder And program from lab 45