Travelling Salesman Problem
Design techniques pseudo codes
Emre Can Kucukoglu
eckucukoglu@gmail.com
https://github.com/eckucukoglu/tsp-solver

1 Backtracking

Algorithm 1 Backtracking design technique for Travelling Salesman Problem

```
procedure TSPBacktracking(level, optcost, optx, n)
    Input: level, n
    Output: optcost, optx

    if lev == n then
        C = cost of x
        if C <optcost then
            optcost = C
            optx = x
        end if
    else
        Compute B = B(x)
        x[level] = 2
        while B <optcost and $x[level] \leq n$ do
            if x[level] is distinct from x[1],...,x[level-1] then
                TSPBacktracking(level+1, optcost, optx, n)
            end if
            $x[level] = x[level] + 1$
        end while
    end if
end procedure
```

2 Branch & Bound

Algorithm 2 Branch & Bound design technique for Travelling Salesman Problem

```
procedure TSPBranchAndBound(level, optcost, optx, n)
    Input: level, n
    Output: optcost, optx
    Var: B, C, Count, NextCoord, NextB

    if lev == n then
        C = cost of x
        if C <optcost then
            optcost = C
            optx = x
        end if
    else
        Count = 0
        for x[level] = 2 to n do
            if x[level] is distinct from x[1],...,x[level-1] then
                Count = Count + 1
                NextCoord[Count] = x[level]
                NextB[Count] = B(x)
            end if
        end for
        Sort NextCoord according to NextB values
        Count = 1
        while $Count \leq n - level$ and NextB[count] <optcost] do
            if x[level] is distinct from x[1],...,x[level-1] then
                x[level] = NextCoord[Count]
                TSPBranchAndBound(level+1, optcost, optx, n)
            end if
            Count = Count + 1
        end while
    end if
end procedure
```