COT 6405
ANALYSIS OF ALGORITHMS

Traveling Salesman Problem
Branch-and-Bound Algorithm

Computer & Electrical Engineering and Computer Science Department
Florida Atlantic University

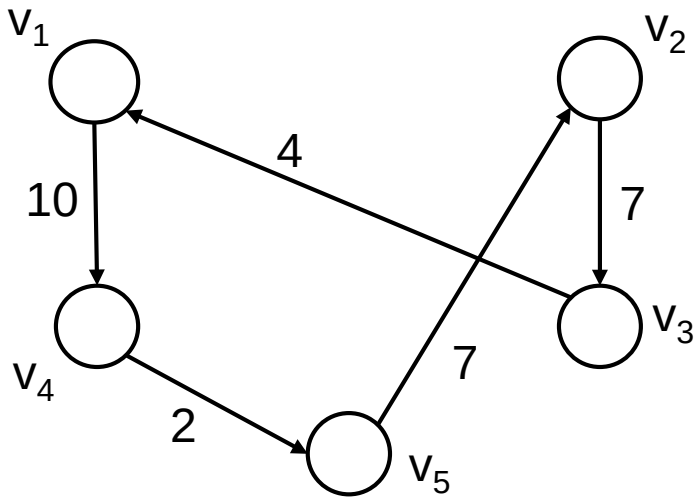
Traveling Salesman Problem (TSP)

Given *n* cities with known distances between each pair, find the shortest tour that passes through all the cities exactly once before returning to the starting city

- Example:

Adjacency matrix:

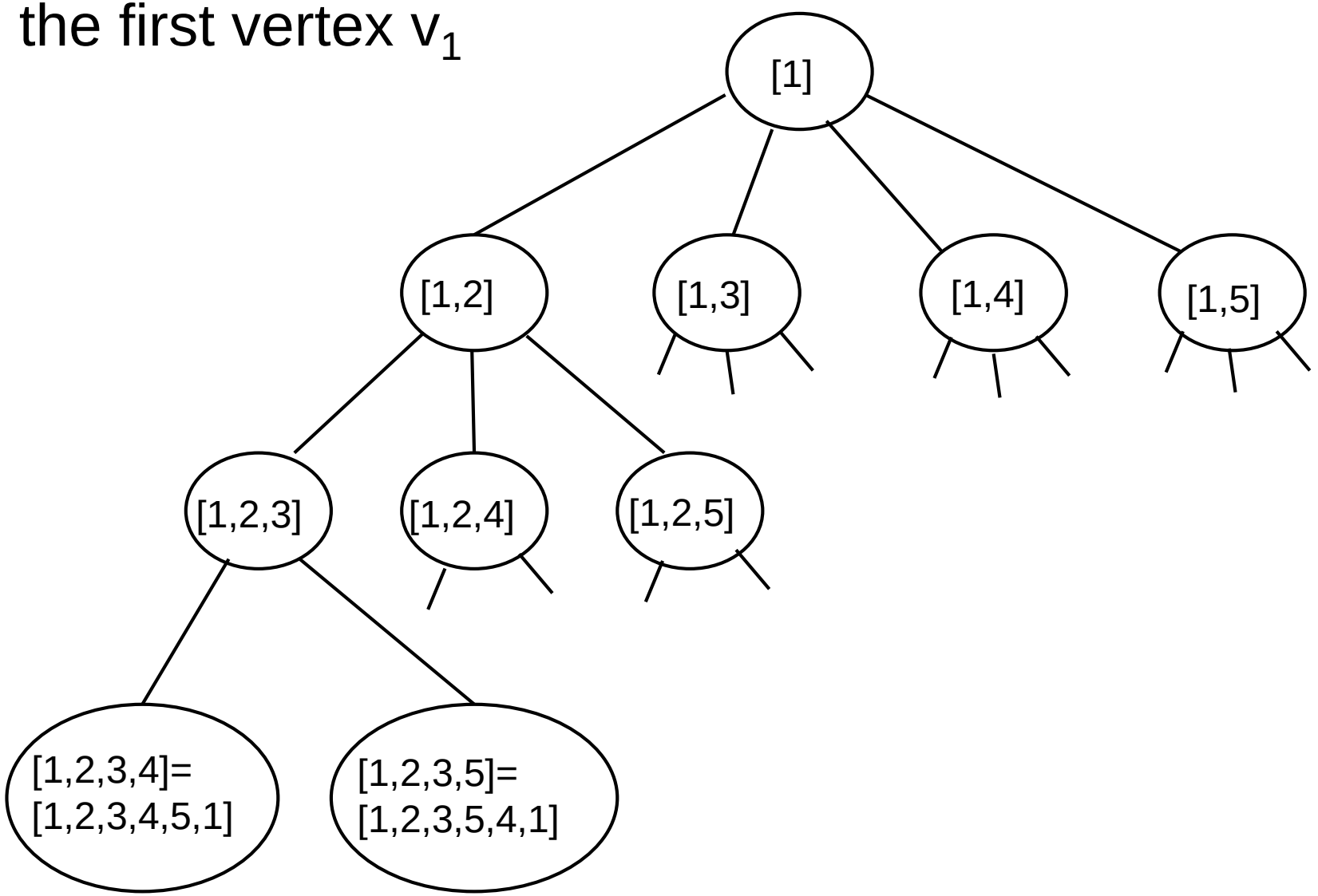
$$\begin{bmatrix} 0 & 14 & 4 & 10 & 20 \\ 14 & 0 & 7 & 8 & 7 \\ 4 & 5 & 0 & 7 & 16 \\ 11 & 7 & 9 & 0 & 2 \\ 18 & 7 & 17 & 4 & 0 \end{bmatrix}$$



min tour length = 30

TSP – search tree example

- Since the tour passes through all vertices, assume that it starts from the first vertex v_1



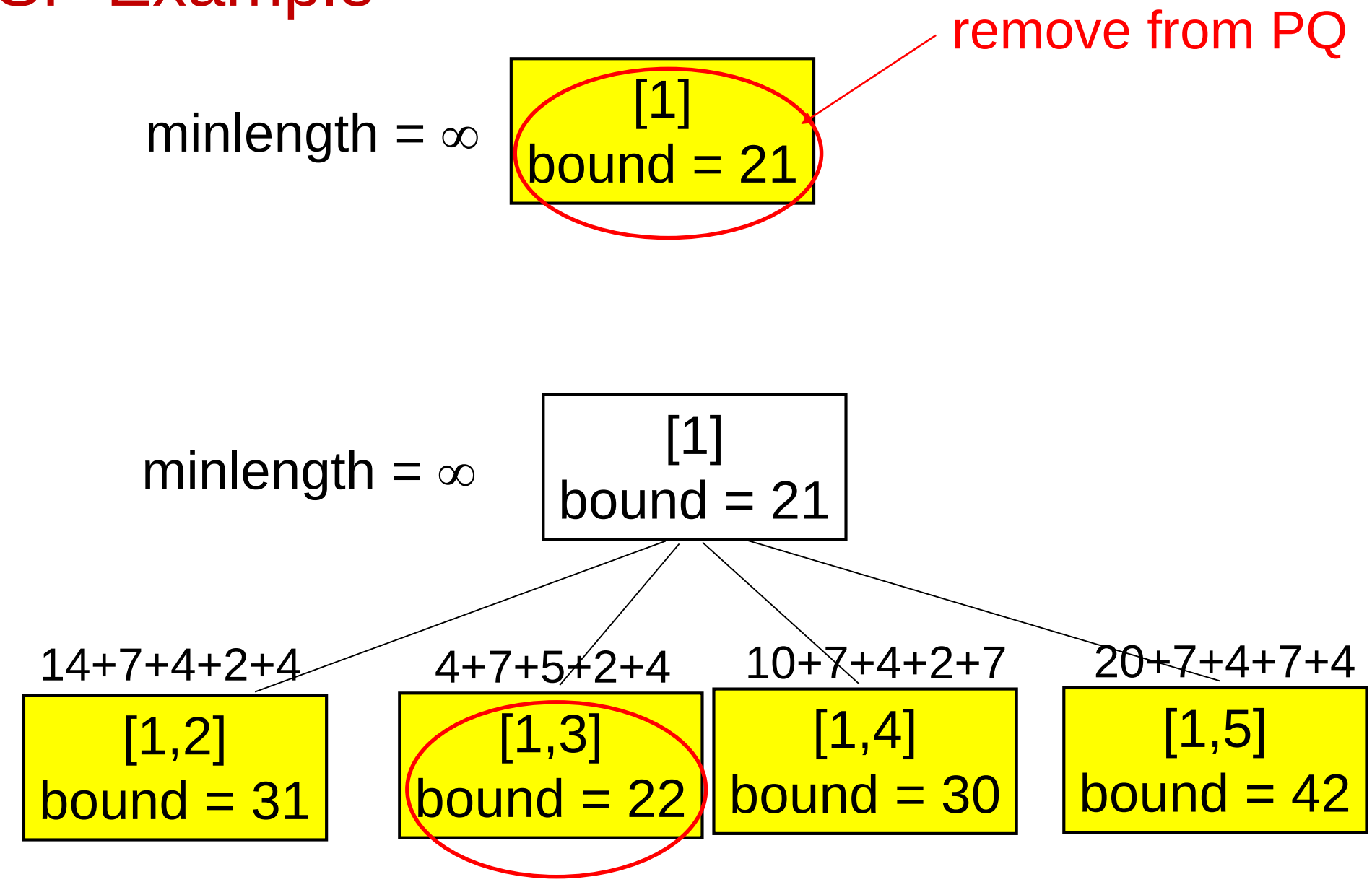
TSP with BestFS with Branch-and-Bound pruning

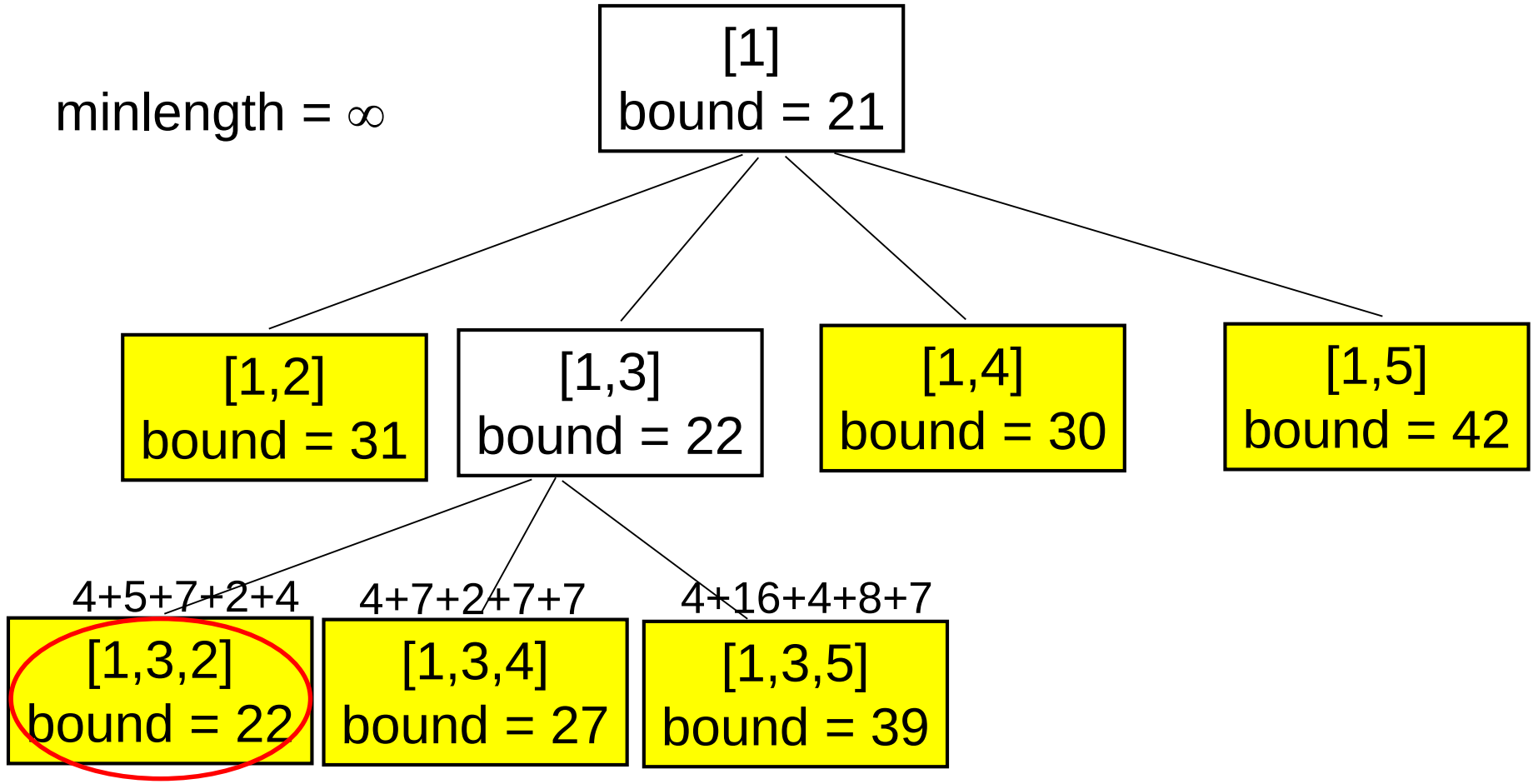
- compute a bound for each node
- lower bound on the length of any tour that can be obtained by expanding beyond a given node
- any tour must leave each vertex exactly once, then a lower-bound is to take minimum edge leaving every vertex:

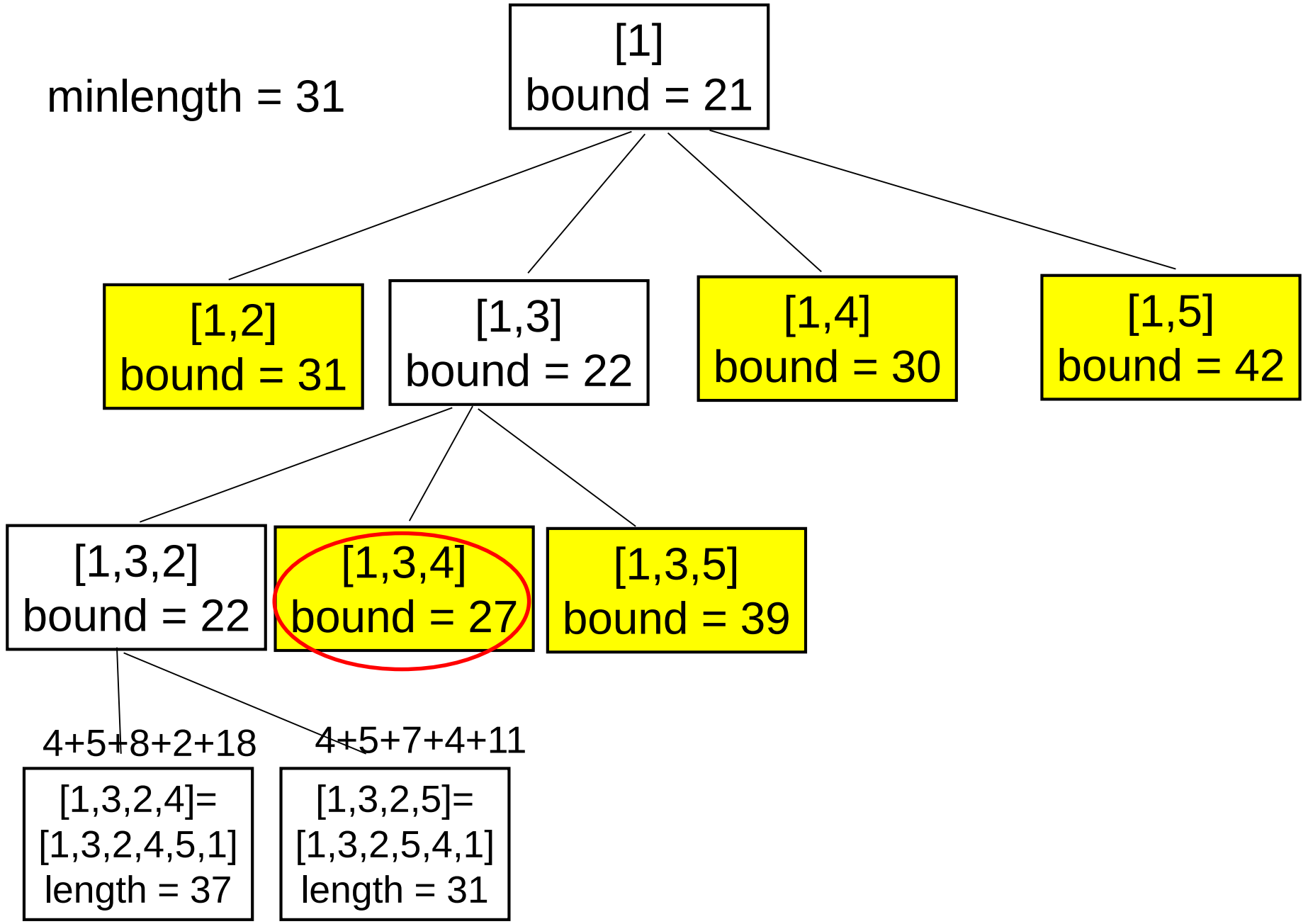
  v_1 minimum(14,4,10,20) = 4
  v_2 minimum(14,7,8,7) = 7
  v_3 minimum(4,5,7,16) = 4
  v_4 minimum(11,7,9,2) = 2
  v_5 minimum(18,7,17,4) = 4

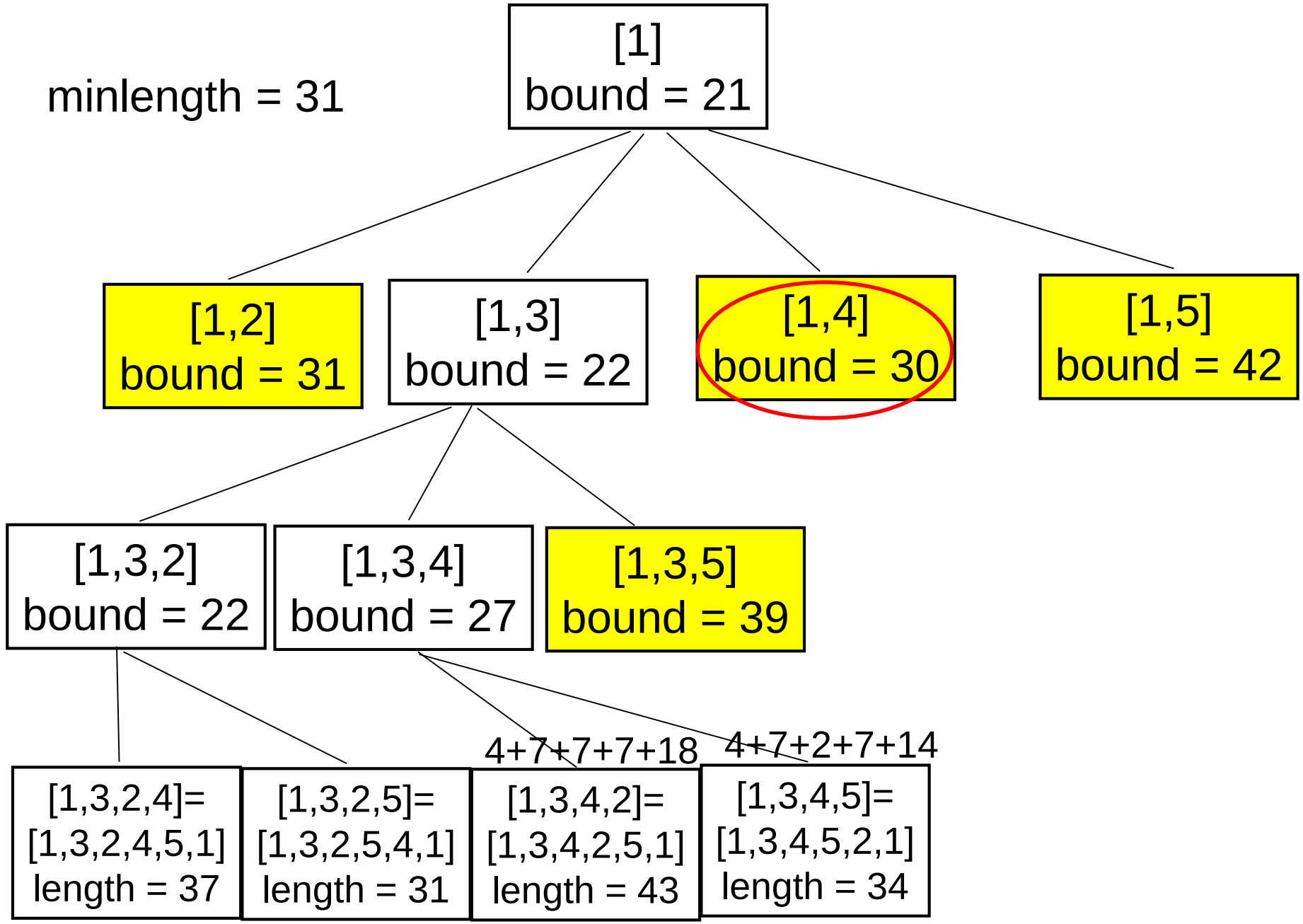
- lower-bound on the length of a tour is 4+7+4+2+4 = 21
- observation: this does not mean there is a tour with this length; it means there is no tour with a shorter length

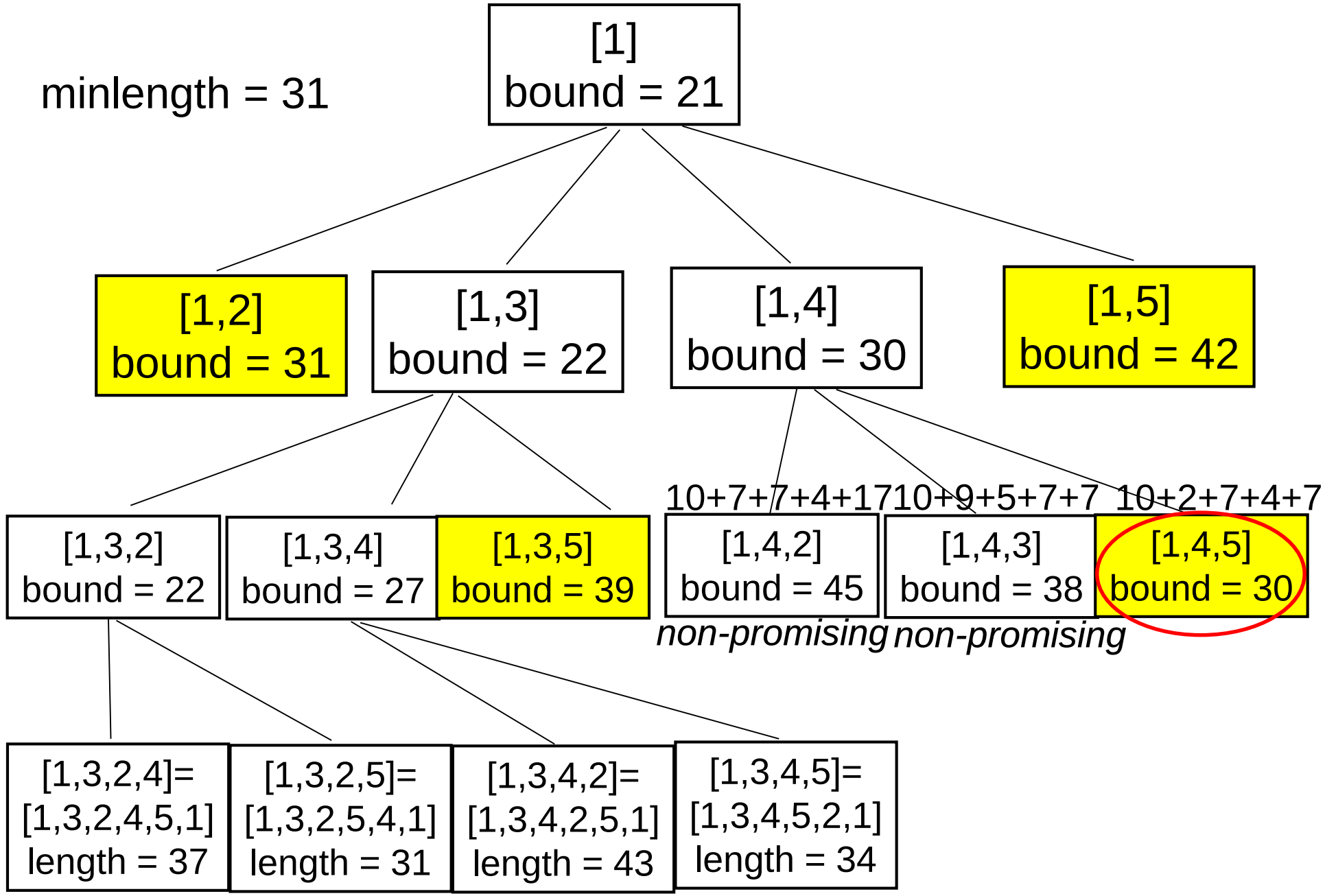
TSP Example

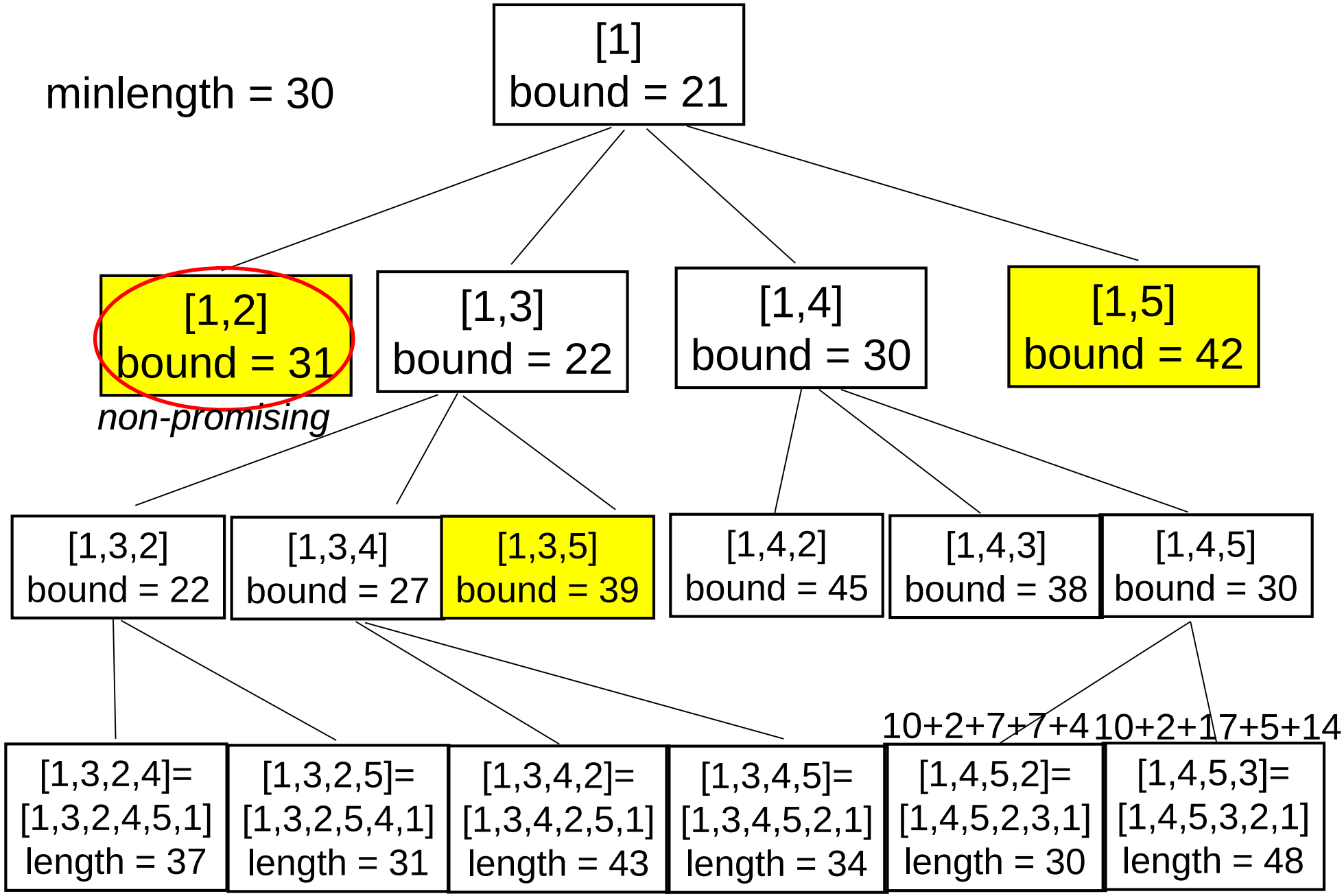

remove from PQ
minlength = ∞
[1]
bound = 21
minlength = ∞
[1]
bound = 21
14+7+4+2+4
4+7+5+2+4
10+7+4+2+7
20+7+4+7+4
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42

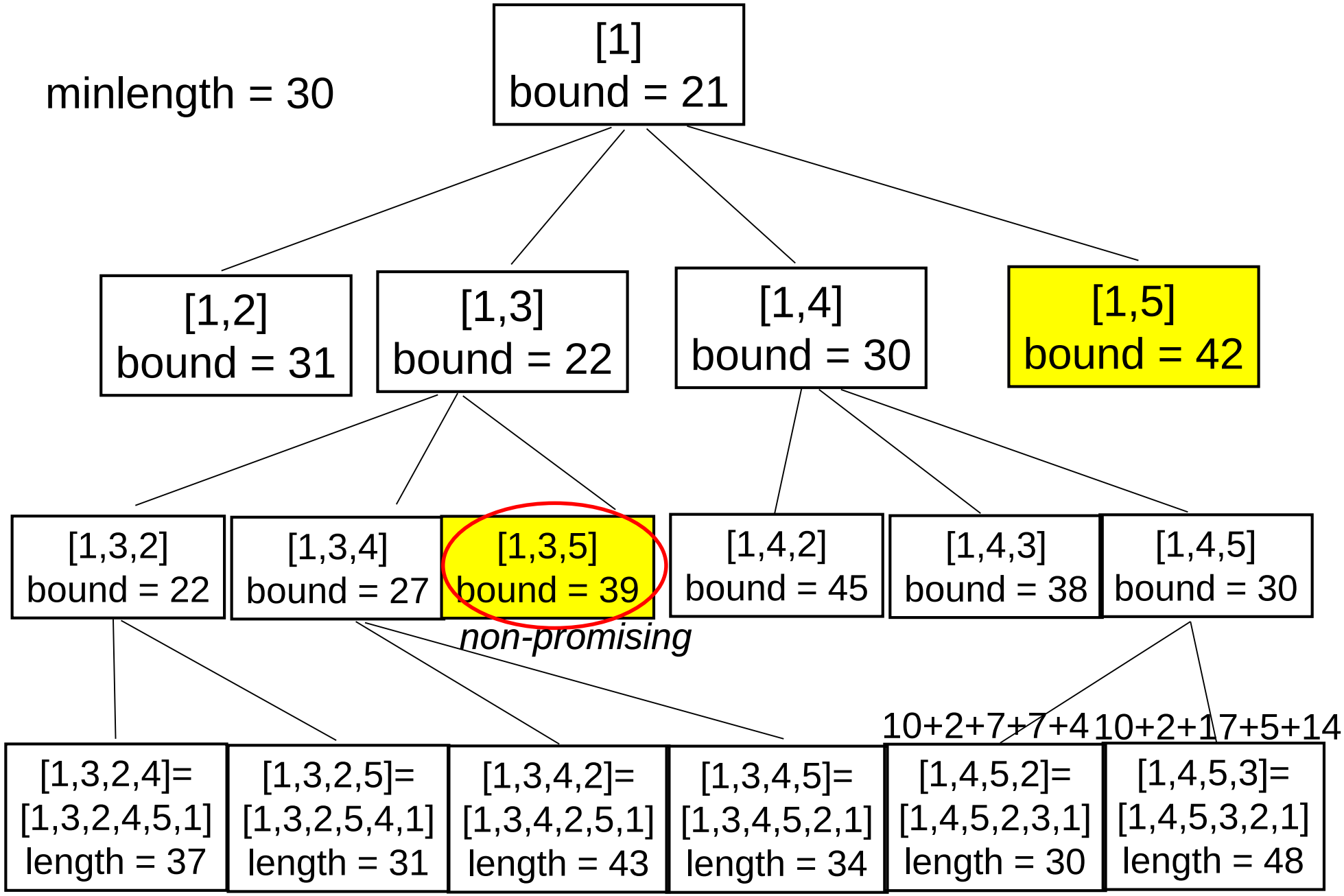

minlength = ∞
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
4+5+7+2+4
4+7+2+7+7
4+16+4+8+7
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39

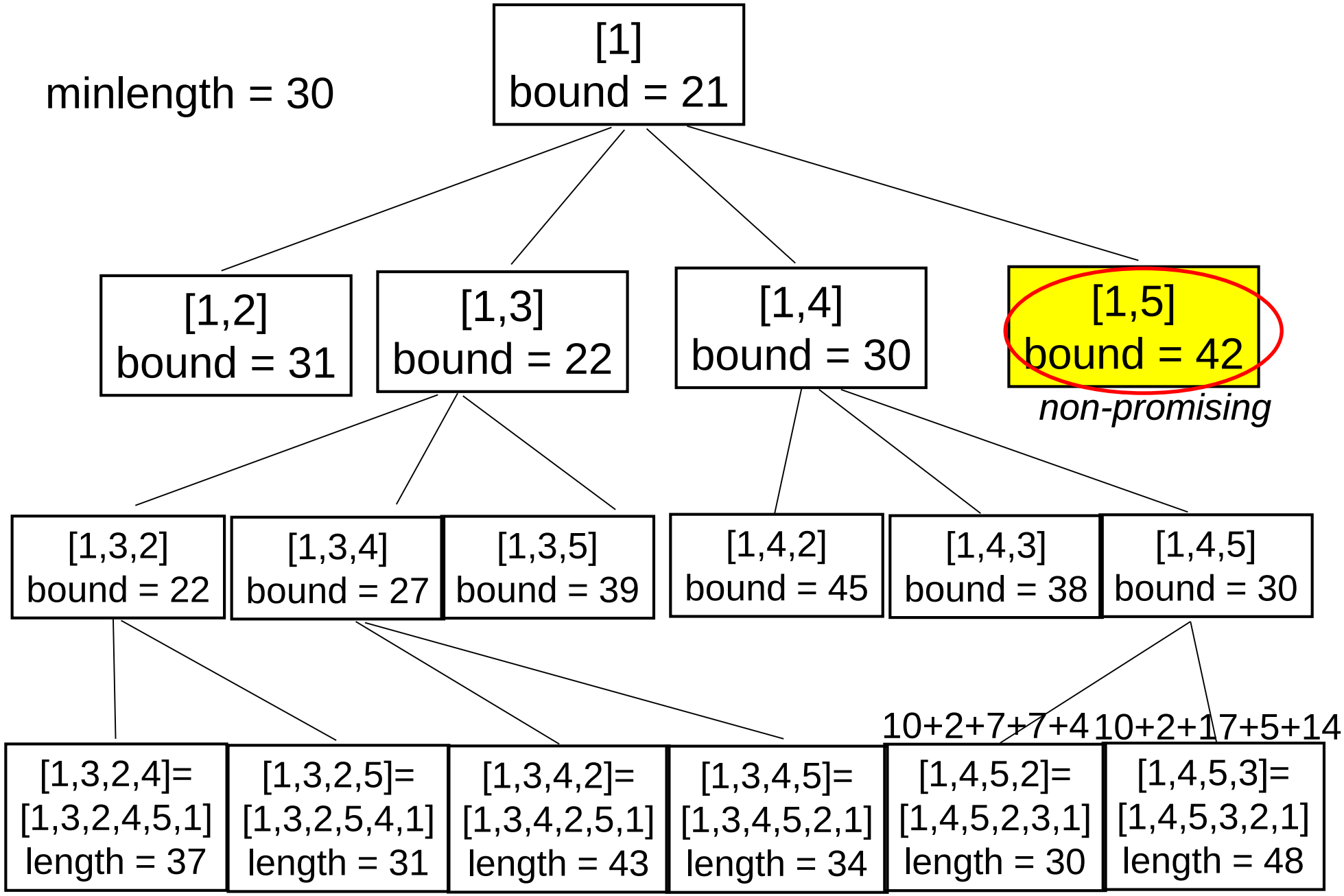

minlength = 31
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
4+5+8+2+18
4+5+7+4+11
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31

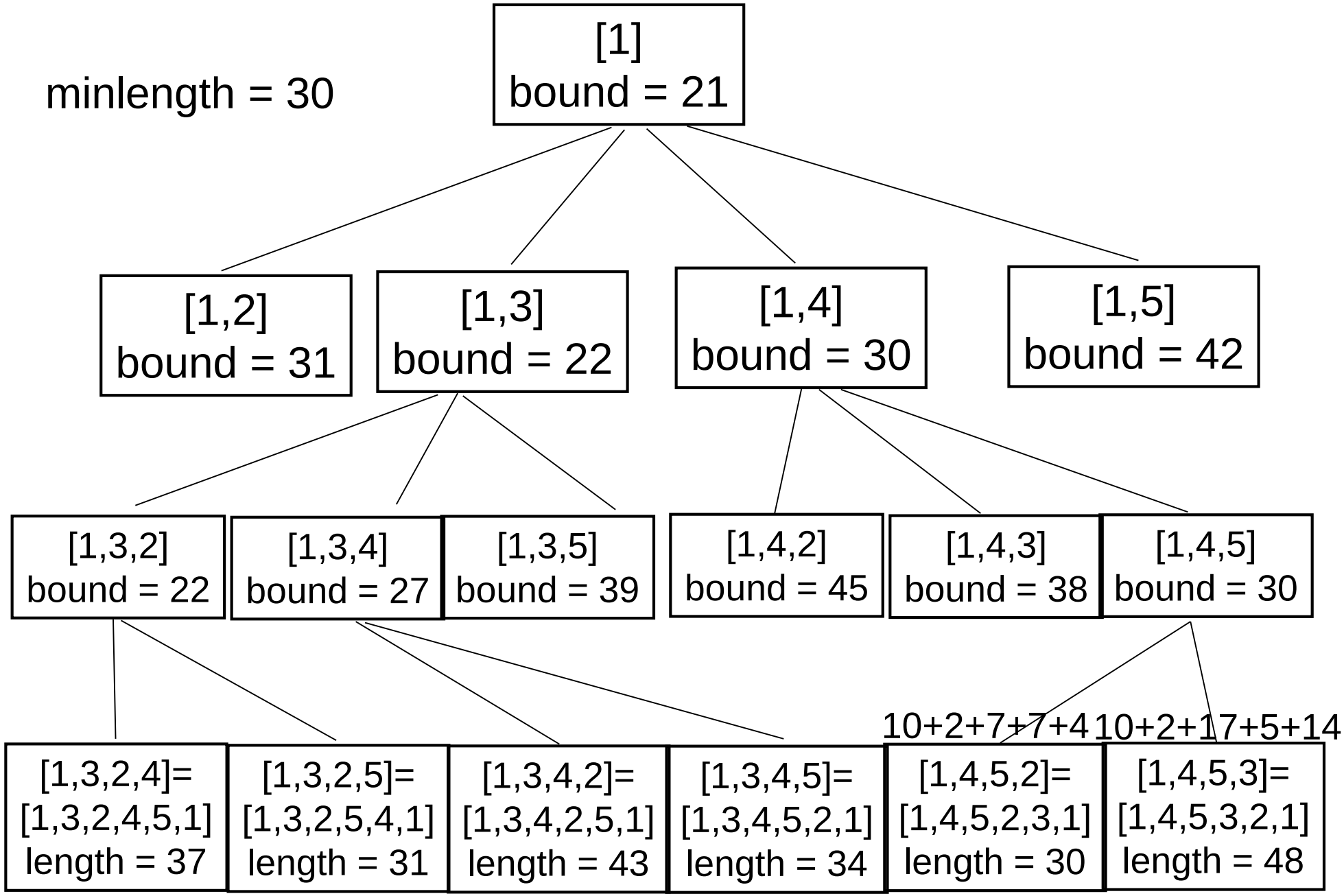

minlength = 31
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
4+7+7+7+18
4+7+2+7+14
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31
[1,3,4,2]=
[1,3,4,2,5,1]
length = 43
[1,3,4,5]=
[1,3,4,5,2,1]
length = 34


minlength = 31
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
10+7+7+4+17
10+9+5+7+7
10+2+7+4+7
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
[1,4,2]
bound = 45
[1,4,3]
bound = 38
[1,4,5]
bound = 30
non-promising
non-promising
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31
[1,3,4,2]=
[1,3,4,2,5,1]
length = 43
[1,3,4,5]=
[1,3,4,5,2,1]
length = 34


minlength = 30
[1]
bound = 21
[1,2]
bound = 31
non-promising
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
[1,4,2]
bound = 45
[1,4,3]
bound = 38
[1,4,5]
bound = 30
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31
[1,3,4,2]=
[1,3,4,2,5,1]
length = 43
[1,3,4,5]=
[1,3,4,5,2,1]
length = 34
10+2+7+7+4
[1,4,5,2]=
[1,4,5,2,3,1]
length = 30
10+2+17+5+14
[1,4,5,3]=
[1,4,5,3,2,1]
length = 48


minlength = 30
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
[1,4,2]
bound = 45
[1,4,3]
bound = 38
[1,4,5]
bound = 30
non-promising
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31
[1,3,4,2]=
[1,3,4,2,5,1]
length = 43
[1,3,4,5]=
[1,3,4,5,2,1]
length = 34
10+2+7+7+4
[1,4,5,2]=
[1,4,5,2,3,1]
length = 30
10+2+17+5+14
[1,4,5,3]=
[1,4,5,3,2,1]
length = 48


minlength = 30
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
non-promising
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
[1,4,2]
bound = 45
[1,4,3]
bound = 38
[1,4,5]
bound = 30
10+2+7+7+4
10+2+17+5+14
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31
[1,3,4,2]=
[1,3,4,2,5,1]
length = 43
[1,3,4,5]=
[1,3,4,5,2,1]
length = 34
[1,4,5,2]=
[1,4,5,2,3,1]
length = 30
[1,4,5,3]=
[1,4,5,3,2,1]
length = 48


minlength = 30
[1]
bound = 21
[1,2]
bound = 31
[1,3]
bound = 22
[1,4]
bound = 30
[1,5]
bound = 42
[1,3,2]
bound = 22
[1,3,4]
bound = 27
[1,3,5]
bound = 39
[1,4,2]
bound = 45
[1,4,3]
bound = 38
[1,4,5]
bound = 30
10+2+7+7+4
10+2+17+5+14
[1,3,2,4]=
[1,3,2,4,5,1]
length = 37
[1,3,2,5]=
[1,3,2,5,4,1]
length = 31
[1,3,4,2]=
[1,3,4,2,5,1]
length = 43
[1,3,4,5]=
[1,3,4,5,2,1]
length = 34
[1,4,5,2]=
[1,4,5,2,3,1]
length = 30
[1,4,5,3]=
[1,4,5,3,2,1]
length = 48

TSP with BestFS with Branch-and-bound pruning

- Each node is an object with fields:

  v.level – node's level in the tree

  v.path – path of the partial tour up to node v

  v.bound – lower bound on the length of any tour expanding beyond this partial tour

TSP with BestFS w/ Branch-and-bound pruning(n,W[ ][ ],opttour, minlength)

```
PQ = ∅
r.level = 0
r.path = [1]
r.bound = bound(r)
minlength = ∞
insert(PQ,r)
while PQ ≠ ∅
    v = remove(PQ)
    if v.bound < minlength
            u.level = v.level+1
            for all i such that 2 ≤ i ≤ n and i is not in v.path
                    u.path = v.path
                    add i at the end of u.path
                    if u.level == n-2     // check if next vertex completes the tour
                            put index of only vertex not in u.path at the end of u.path
                            put 1 at the end of u.path
                            if length(u) < minlength
                                    minlength = length(u)
                                    opttour = u.path
                    else
                            u.bound = bound(u)
                            if u.bound < minlength
                                    insert(PQ,u)
```