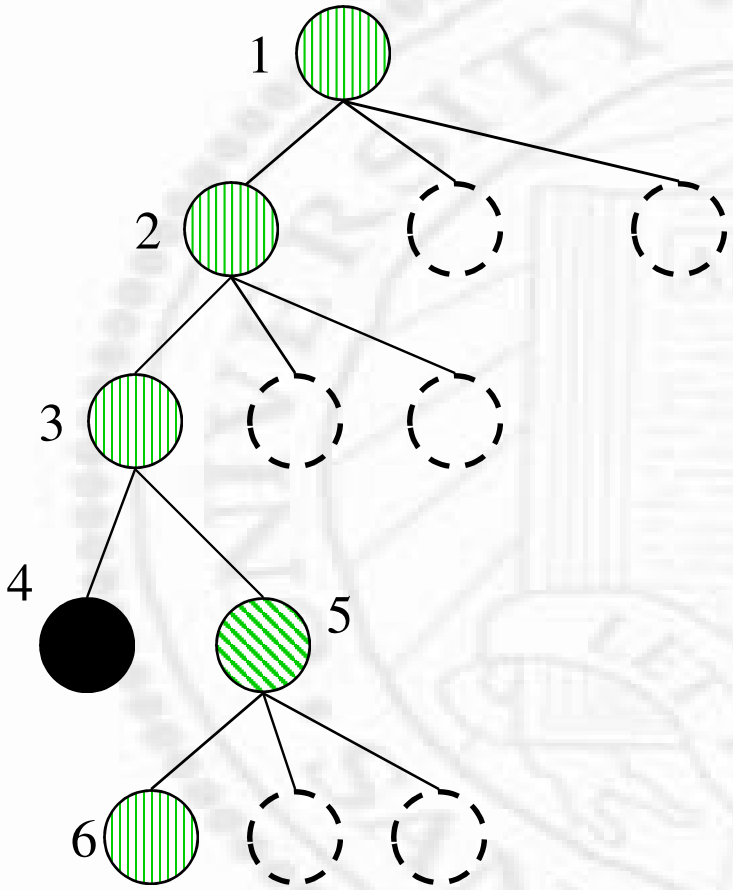
Back Tracking

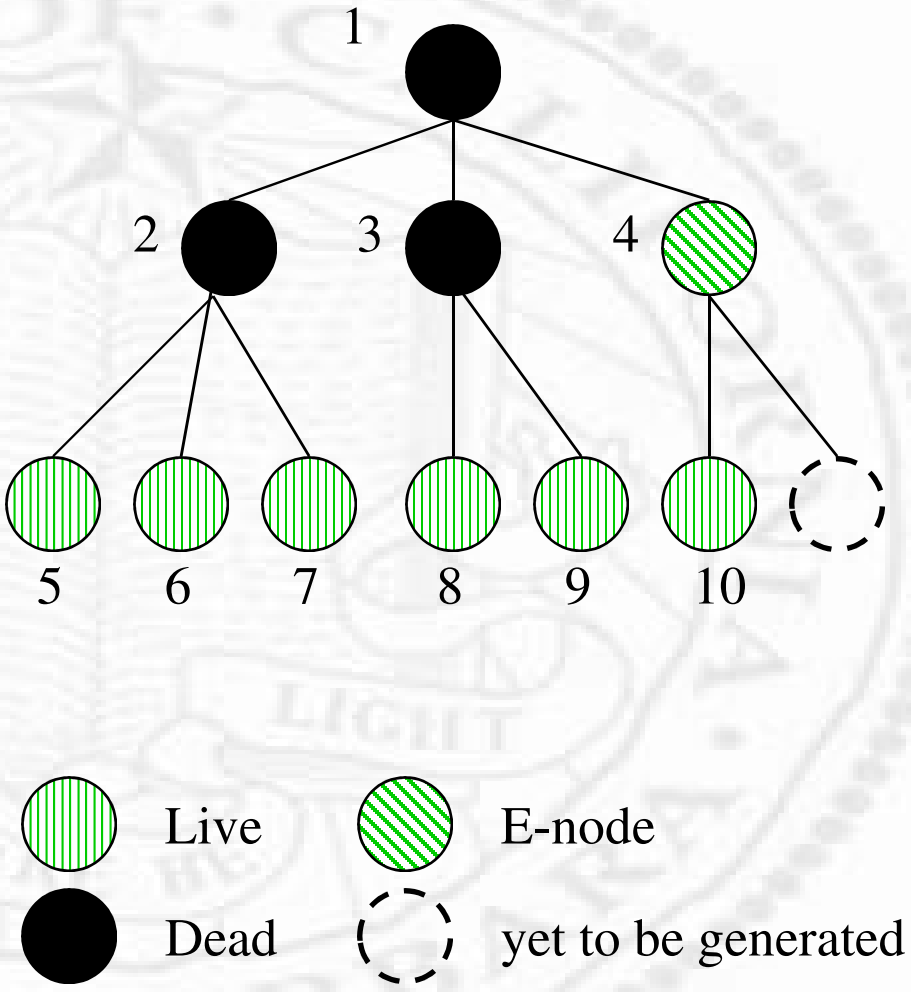
Tree Traversal

- Enumeration (brute force) often results in a tree, for pruning
  - Feasibility (a node by itself)
  - Optimality (comparing a node with others)
  - Merge (DP)
- Look at
  - Different traversal patterns (depth first, breadth first, best first)
  - Bounding (with both feasibility and optimality)

- Depth first



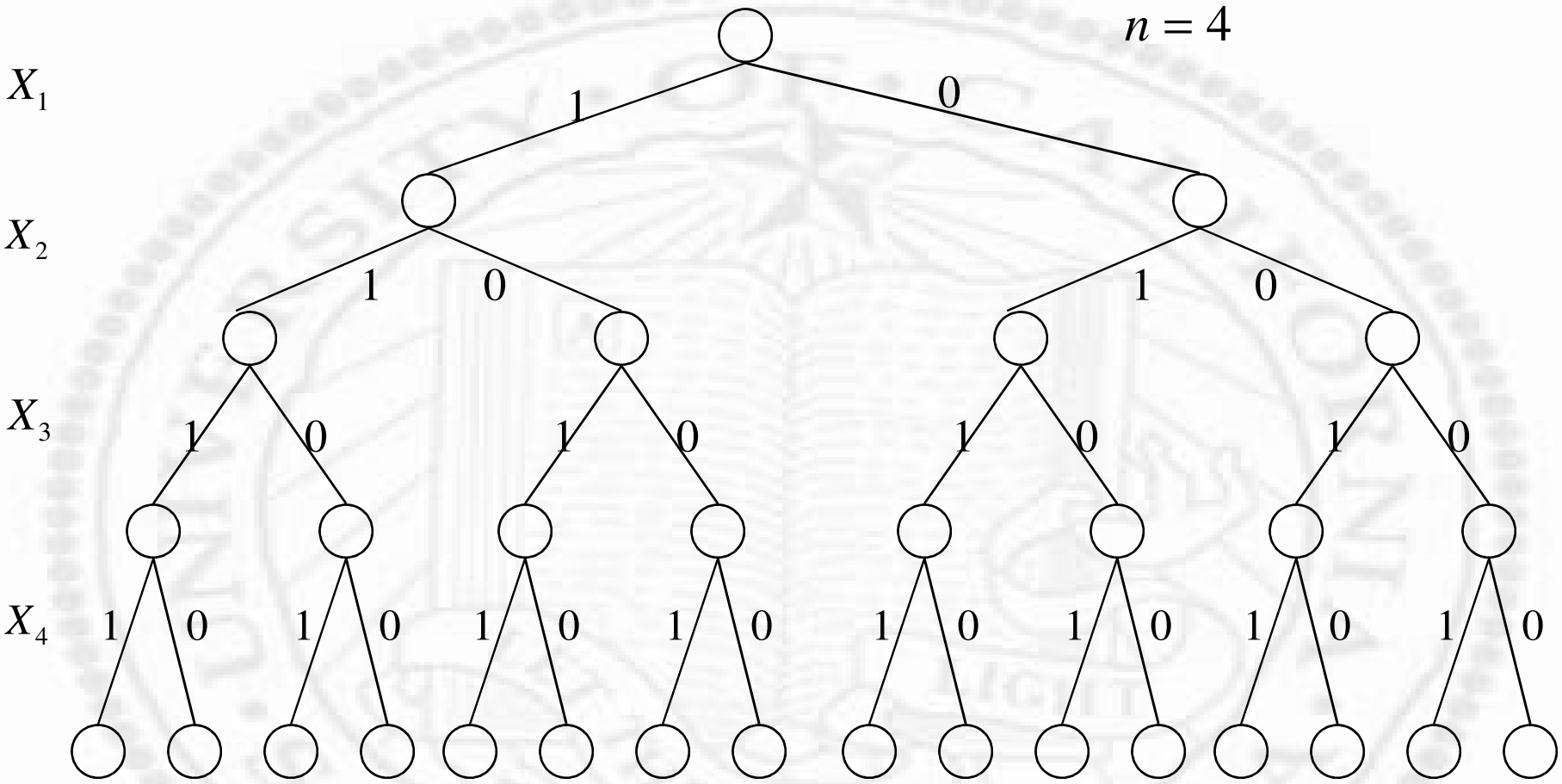
- Breadth first



Sum of Subsets

- Input:
  - a set of n distinct positive numbers
- Output:
  - find all combinations of these numbers which sum up to M
- Explicit constraints: $(X_1, X_2, \ldots, X_n)$, where $X_i = 0$ or 1, $1 \le i \le n$
- Feasibility constraints: $\sum_{1 \le i \le n} W_i X_i = M$

Explicit Constraints



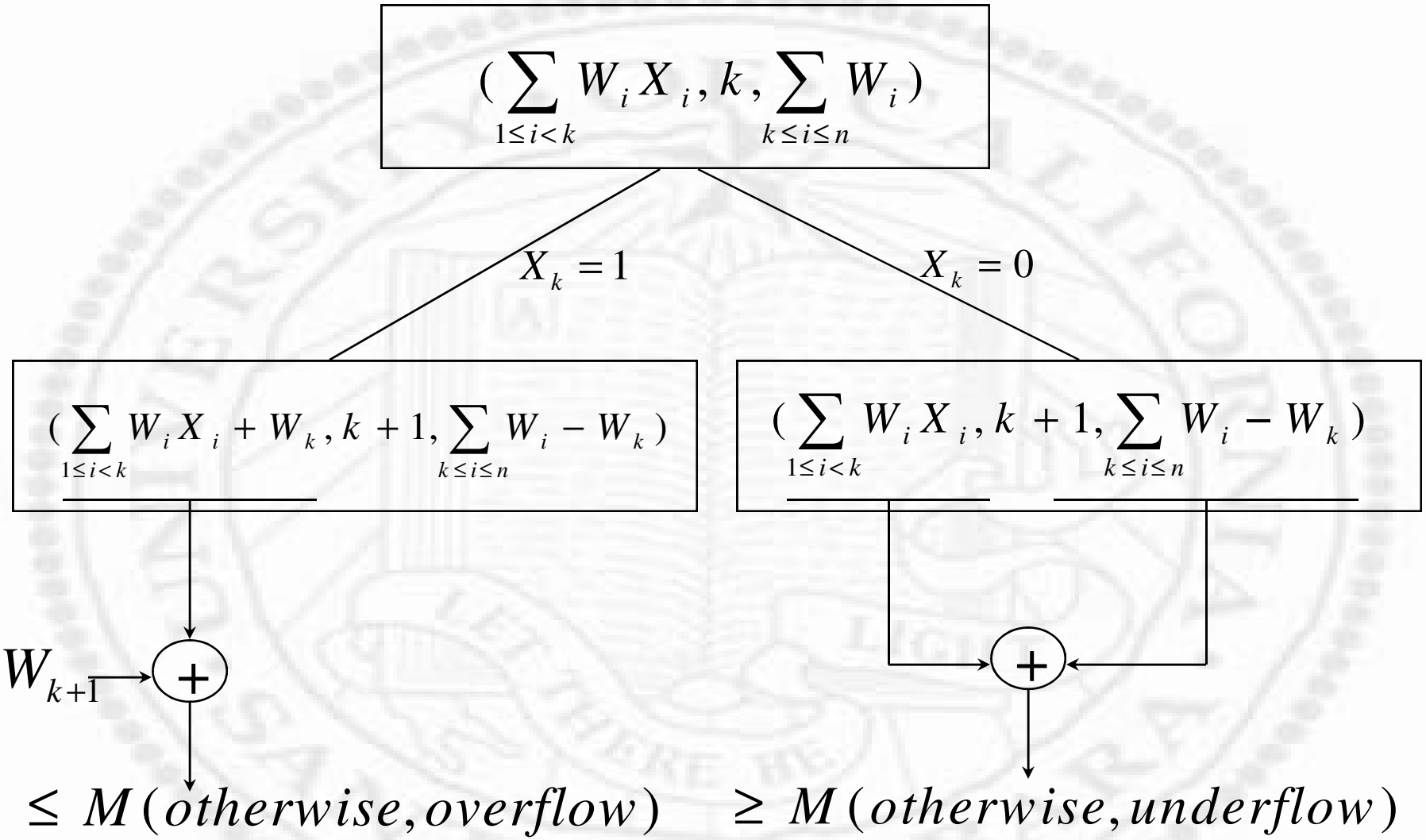
$$Complexity: O(2^n)$$

- Feasibility Constraints: how can a problem state not lead to an answer state?
  - Overflow
  - Underflow

Assume that $W_1 \le W_2 \le \ldots \le W_n$

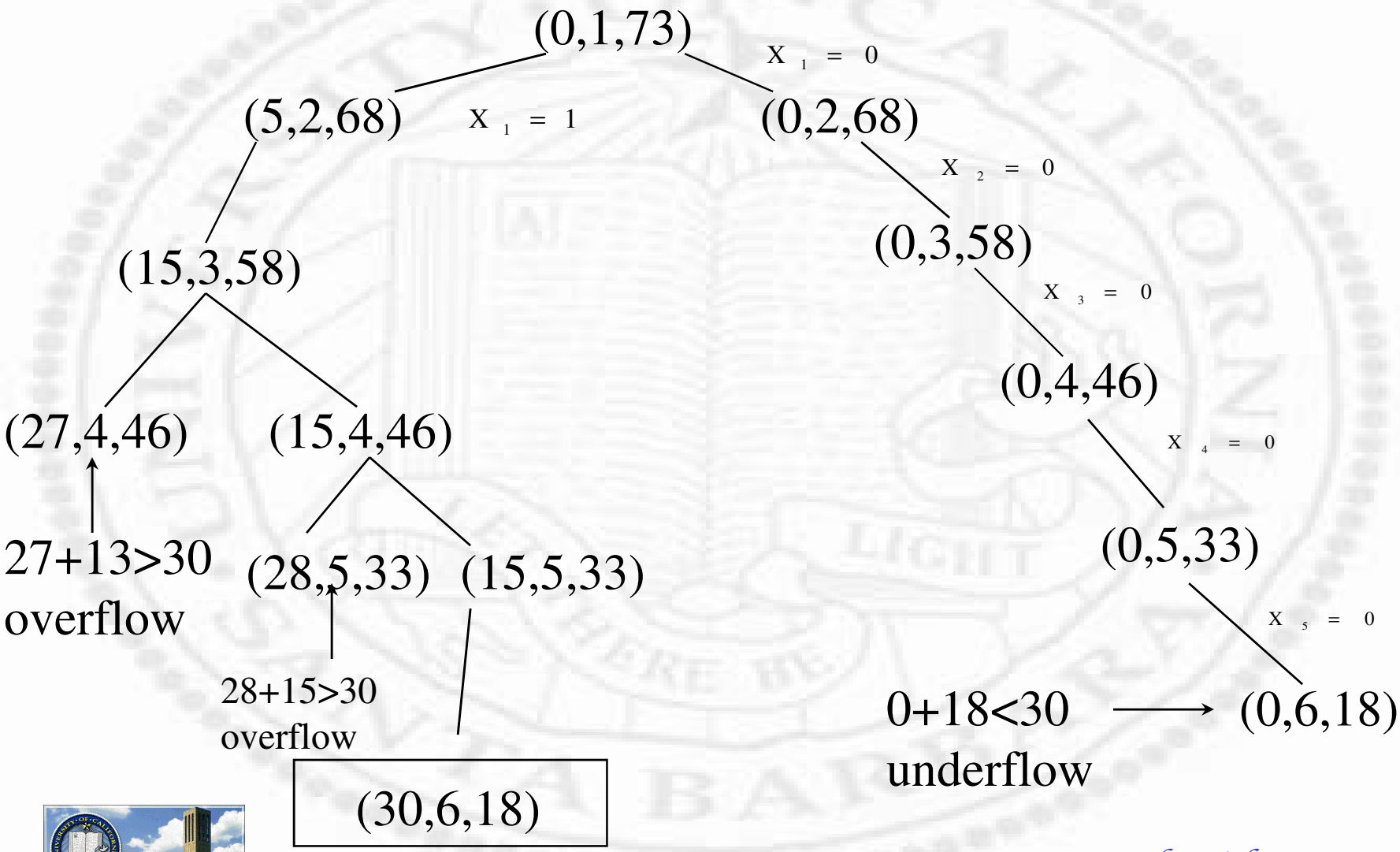
$$\text{Overflow}: \sum_{1 \le i \le k} W_i X_i + W_{k+1} > M$$

$$\text{Underflow}: \sum_{1 \le i \le k} W_i X_i + \sum_{k < i \le n} W_i < M$$


$(\sum_{1\le i<k} W_i X_i, k, \sum_{k\le i\le n} W_i)$
$X_k = 1$
$X_k = 0$
$(\sum_{1\le i<k} W_i X_i + W_k, k+1, \sum_{k\le i\le n} W_i - W_k)$
$(\sum_{1\le i<k} W_i X_i, k+1, \sum_{k\le i\le n} W_i - W_k)$
W_{k+1}
+
+
$\le M(otherwise, overflow)$
$\ge M(otherwise, underflow)$

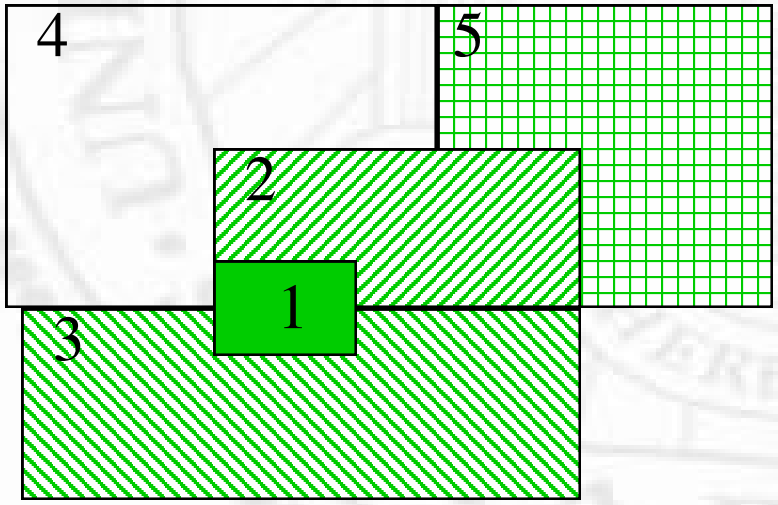
$n = 6,\ M = 30, W_i = (5,10,12,13,15,18)$

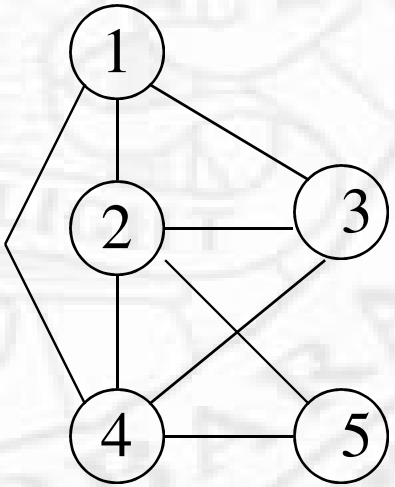
$$\left(\sum_{1 \le i < k} W_i X_i,\ k,\ \sum_{k \le i \le n} W_i\right)$$



Graph Coloring

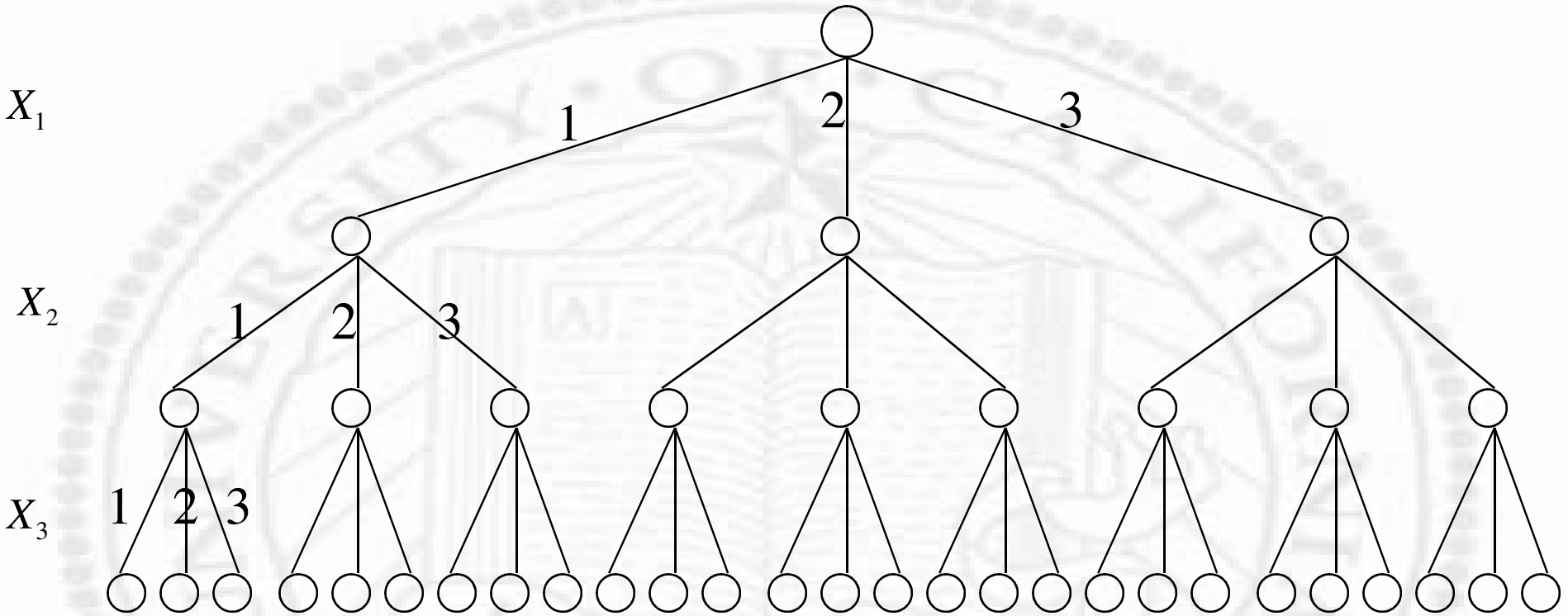
- Input: A planar graph with n nodes
- Output: Use m colors to color the nodes such that no two adjacent nodes share the same color





Explicit Constraints

$n = 4, m = 3$



$$Size: O(m^n)$$

$$Complexity: \sum_{1 \le i \le n} m^i \times (mn) = O(nm^n)$$

0/1-Knapsack

- Explicit constraints:

$$(X_1, X_2, \ldots, X_n), \text{ where } X_i = 0 \text{ or } 1,\ 1 \le i \le n$$

- Feasibility constraints:

$$\sum_{1 \le i \le n} X_i W_i \le M, \sum_{1 \le i \le n} X_i P_i \text{ is maximized}$$

- Bounding function
  - based on *feasibility*

    if $\sum_{1 \le i \le k} X_i W_i > M$ at level k, stop (overflow)

  - based on *optimality*

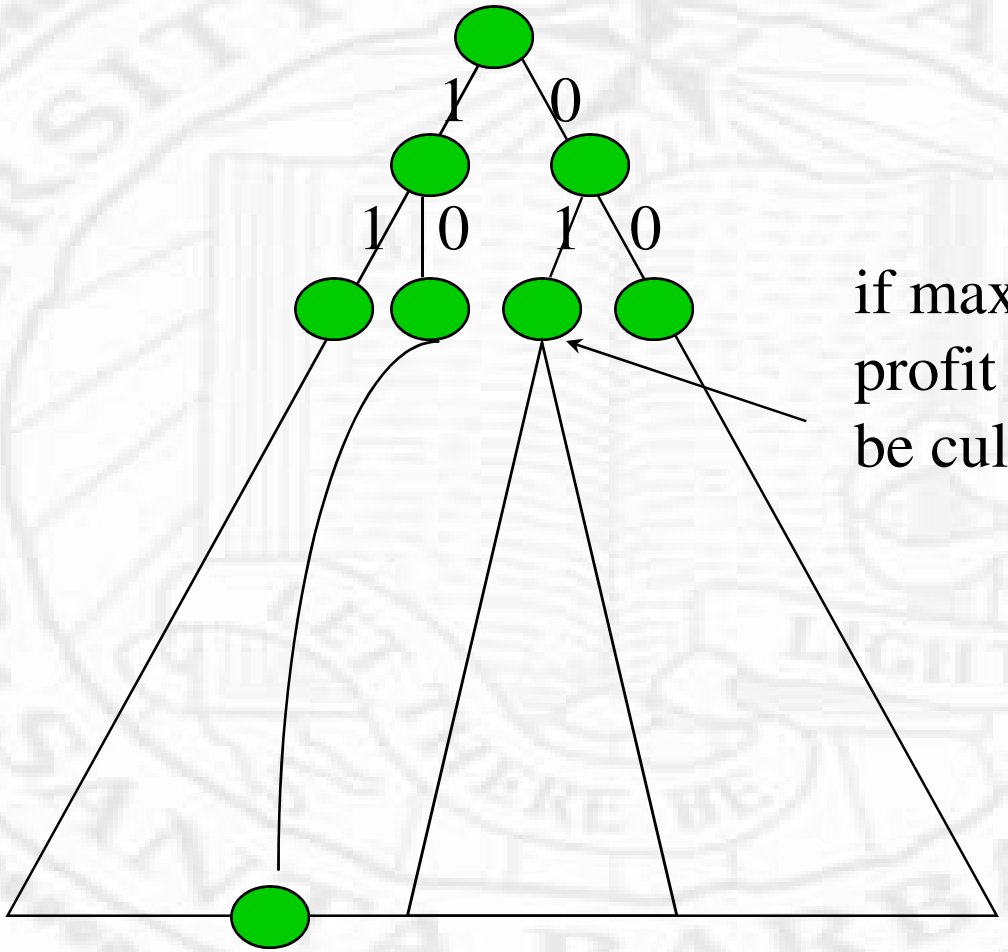
    if at level k, two nodes can be found

    $$\sum_{1 \le i \le k} X_i W_i \le \sum_{1 \le i \le k} X'_i W_i$$

    $$\sum_{1 \le i \le k} X_i P_i \ge \sum_{1 \le i \le k} X'_i P_i$$

    then $(X'_1, X'_2, \ldots, X'_k)$ cannot lead to the optimal solution

- Bounding function
  - based on *traversal pattern with prediction*

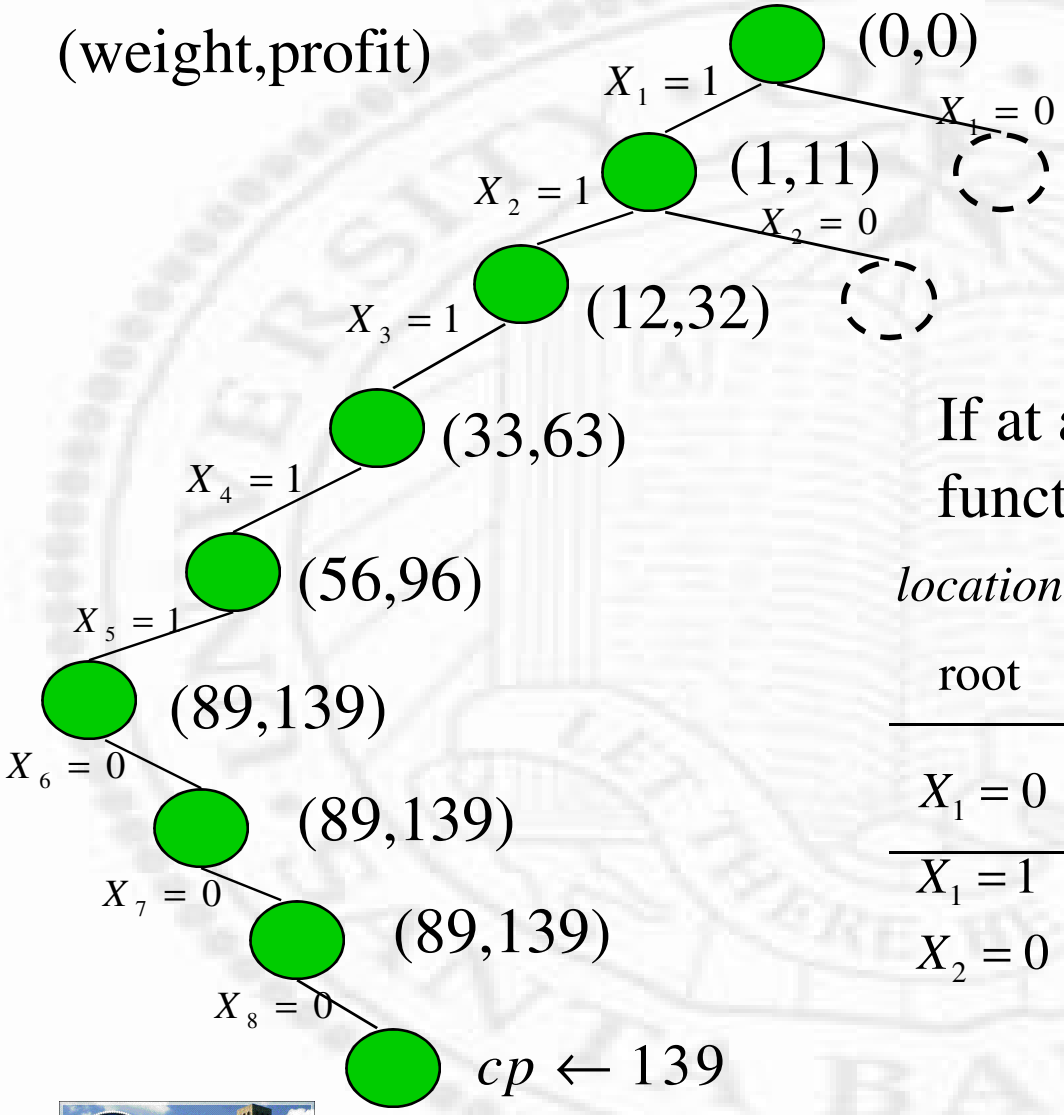


if maximum achievable profit is less than *cp*, can be culled

$$cp \leftarrow \sum_{1 \le i \le n} X_i P_i$$

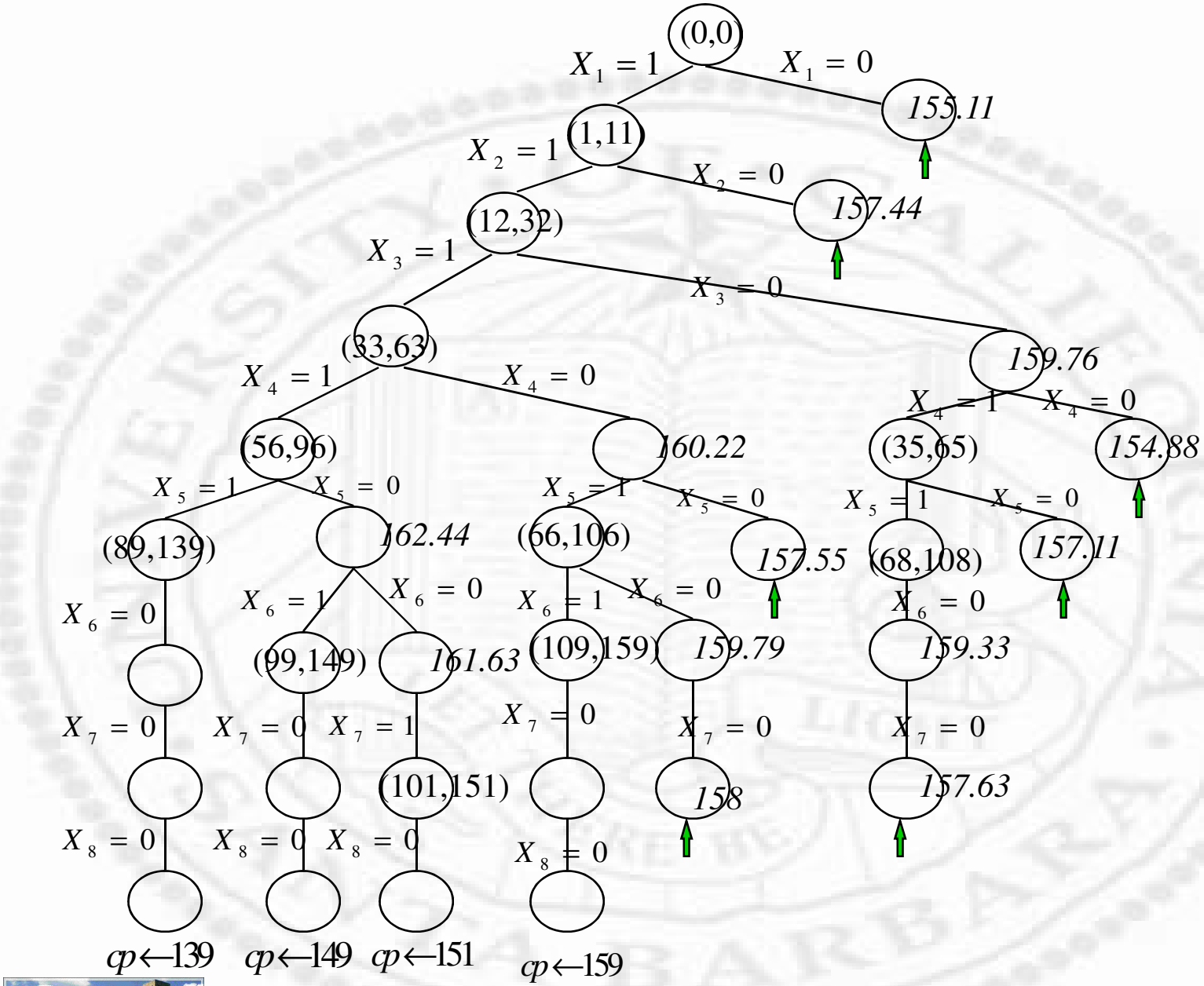
- What is the maximum achievable profit from a node?
  - Include all remaining objects in the sack
    - May not be feasible
  - Include all objects without going over the capacity, but what is the best way to do it?
    - Greedy method
    - Greedy method fills up the knapsack
    - The profit/volume is as high as possible

n=8, W=(1, 11,21,23,33,43,45,55)
M=110,P=(11,21,31,33,43,53,55,65)



If at any node, the bounding function < 139, stop

location	*solution*	max profit
root	$(1,1,1,1,1,\frac{21}{43},0,0)$	164.88
$X_1 = 0$	$(0,1,1,1,1,\frac{22}{43},0,0)$	155.12
$X_1 = 1$ $X_2 = 0$	$(1,0,1,1,1,\frac{32}{43},0,0)$	157.44


(0,0)
$X_1 = 1$
$X_1 = 0$
155.11
(1,11)
$X_2 = 1$
$X_2 = 0$
157.44
(12,32)
$X_3 = 1$
$X_3 = 0$
(33,63)
159.76
$X_4 = 1$
$X_4 = 0$
(56,96)
160.22
(35,65)
154.88
$X_5 = 1$
$X_5 = 0$
(89,139)
162.44
(66,106)
157.55
(68,108)
157.11
$X_6 = 0$
$X_6 = 1$
$X_6 = 0$
(99,149)
161.63
(109,159)
159.79
159.33
$X_7 = 0$
$X_7 = 0$
$X_7 = 1$
(101,151)
158
157.63
$X_8 = 0$
$X_8 = 0$
$X_8 = 0$
cp←139
cp←149
cp←151
cp←159


University of California
Santa Barbara

Bounding Functions

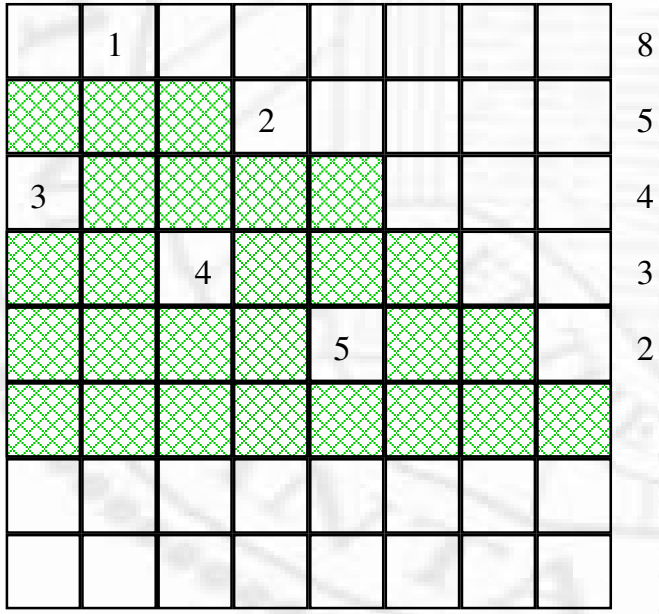
- Based on feasibility (solution's own merit)
  - sum of subsets
  - graph coloring
- Based on optimality (compared with others)
  - 0/1 knapsack
- Based on traversal pattern (with prediction)
  - 0/1 knapsack

Time Complexity

- Explicit constraints $O(m^n)$
  - m choices at each node
  - n decisions to be made
- Implicit constraints
  - complexity should reduce
  - how much less?
  - problem instance dependent

Monte Carlo Method

- Sampling technique
- Select several random instances and average the complexity figures



$$1 + 8 + 8 \times 5 + 8 \times 5 \times 4 + 8 \times 5 \times 4 \times 3 + 8 \times 5 \times 4 \times 3 \times 2 = 1649$$

Only about 2.34% of the total state space tree