Discrete Optimization

The Knapsack Problem: Part II

Goals of the Lecture

- Introduce branch and bound

One-Dimensional Knapsack

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & x_i \in \{0, 1\} \quad (i \in 1..3) \end{array}$$

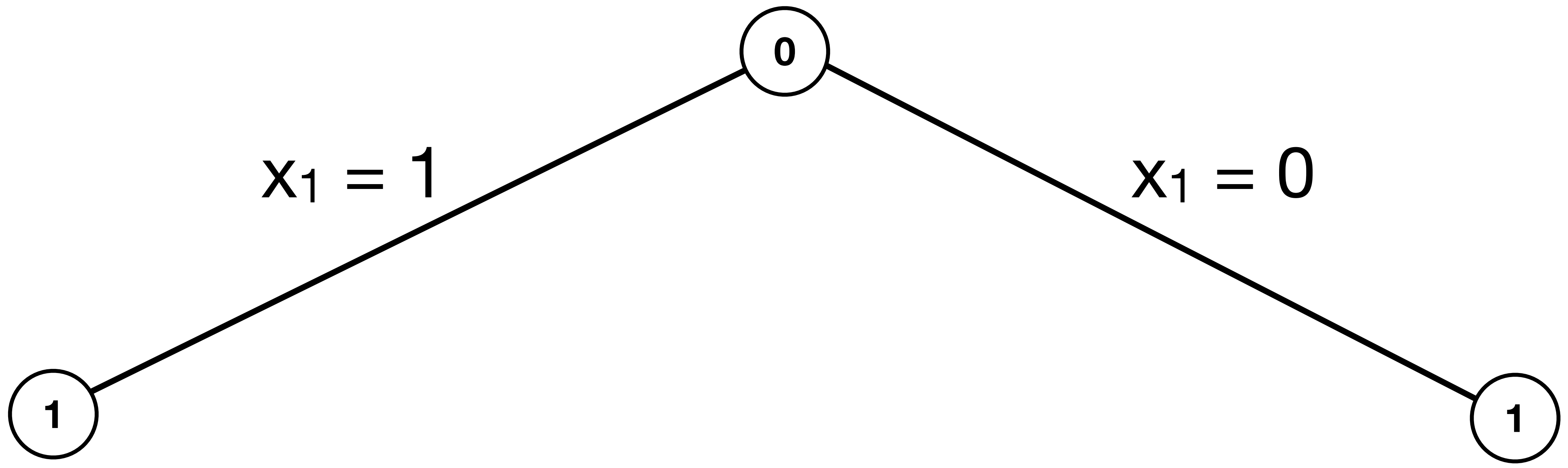
Exhaustive Search

x_1	x_2	x_3
?	?	?

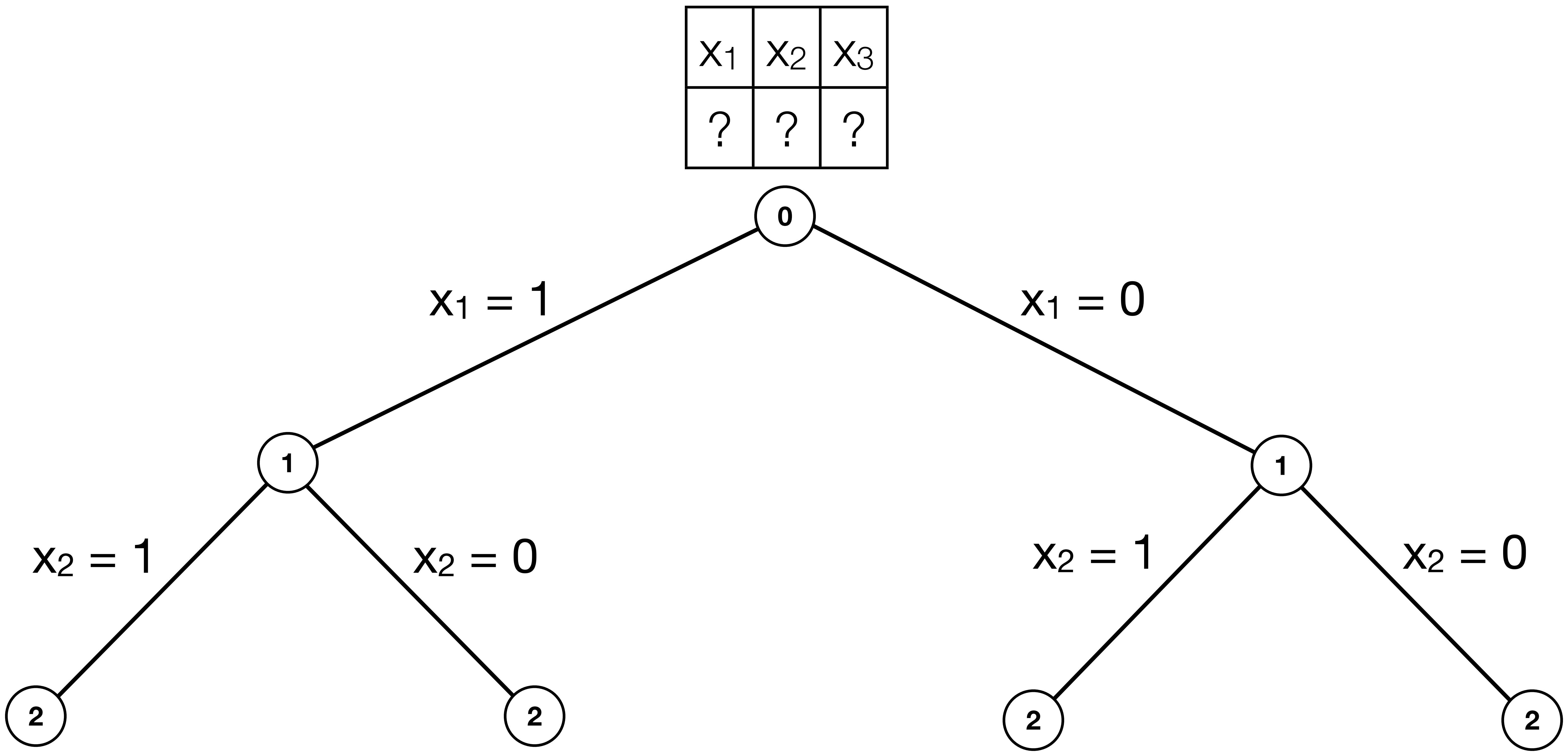
0

Exhaustive Search

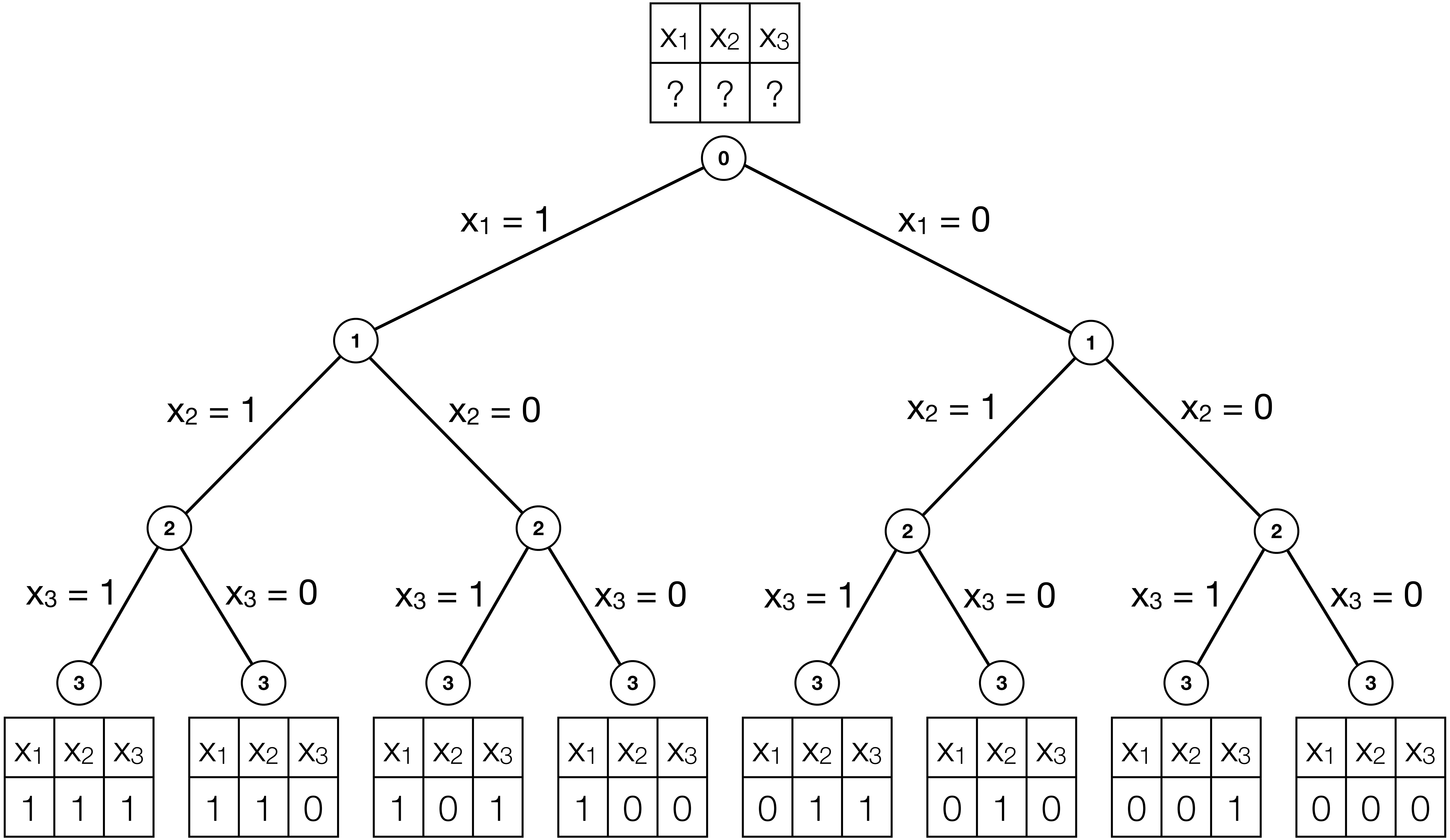
x_1	x_2	x_3
?	?	?



Exhaustive Search



Exhaustive Search



Branch and Bound

- Iterative two steps
  - branching
  - bounding

Branch and Bound

- Iterative two steps
  - branching
  - bounding
- Branching
  - split the problem into a number of subproblems
    - like in exhaustive search

Branch and Bound

- Iterative two steps
  - branching
  - bounding
- Branching
  - split the problem into a number of subproblems
    - like in exhaustive search
- Bounding
  - find an ***optimistic estimate*** of the best solution to the subproblem
    - maximization: upper bound
    - minimization: lower bound

Branch and Bound

- How to find this optimistic estimate?

Branch and Bound

- How to find this optimistic estimate?
  - Relaxation!

Branch and Bound

- How to find this optimistic estimate?
  - Relaxation!

- Optimization is the art of relaxation

A Knapsack Model

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
 & 5x_1 + 8x_2 + 3x_3 \leq 10 \\
 & x_i \in \{0, 1\} \quad (i \in 1..3)
\end{array}
$$

A Knapsack Model

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & x_i \in \{0,1\} \quad (i \in 1..3) \end{array}$$

- What can we relax?

A Knapsack Model

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & x_i \in \{0, 1\} \quad (i \in 1..3) \end{array}$$

- What can we relax?
  - we can relax the capacity constraint

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

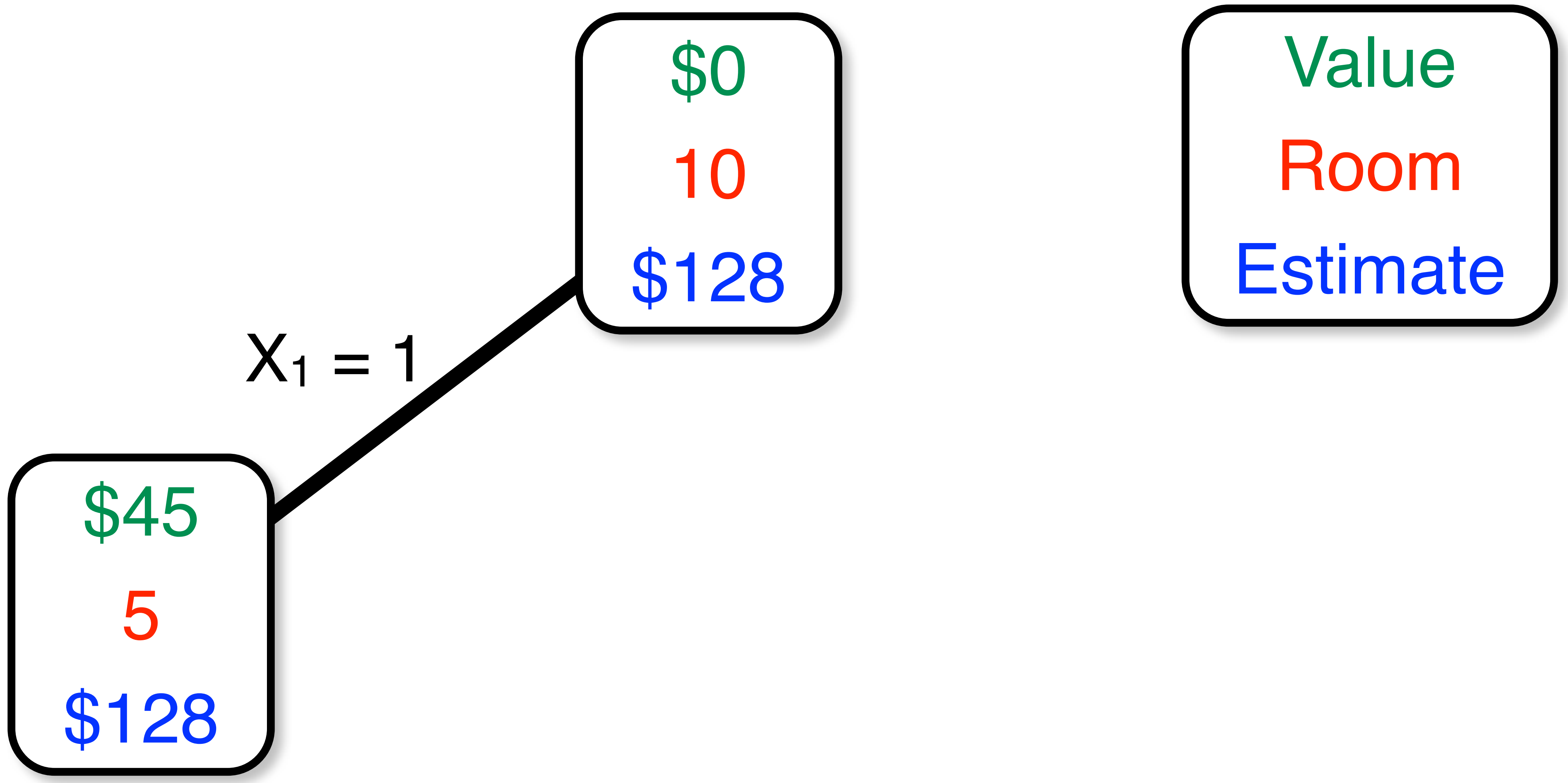
$0
10
$128

Value
Room
Estimate

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

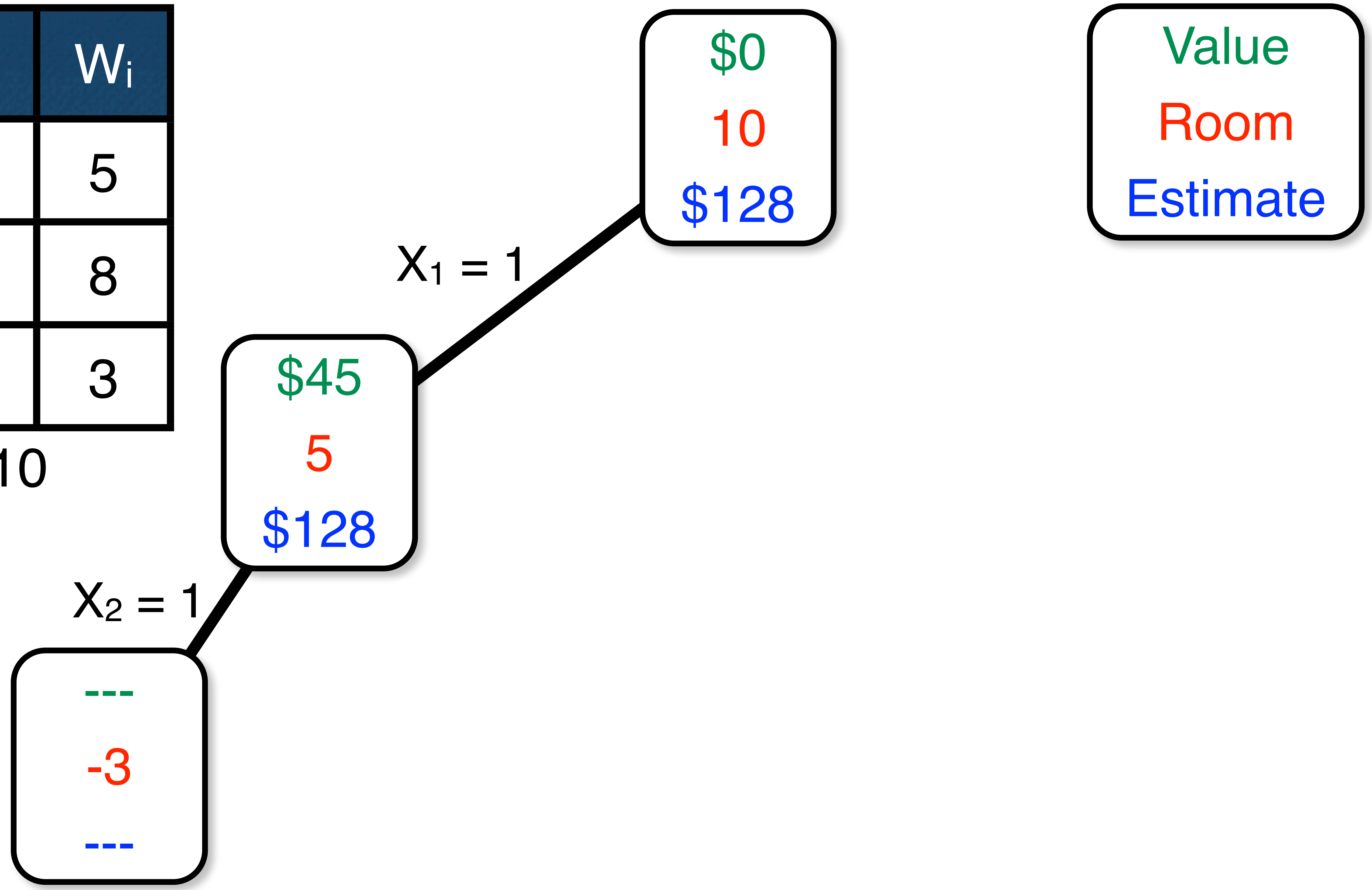
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

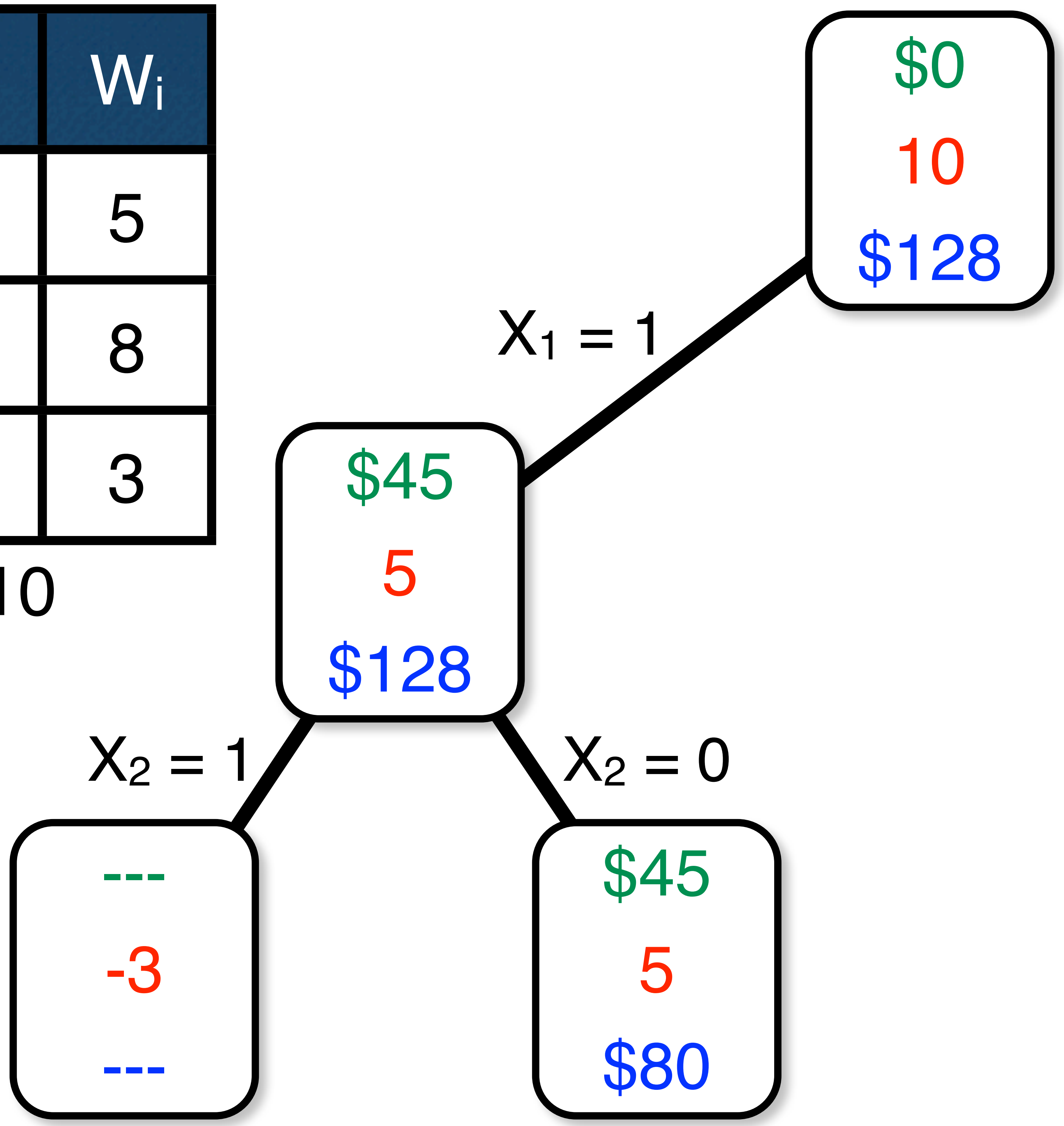
$K = 10$



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

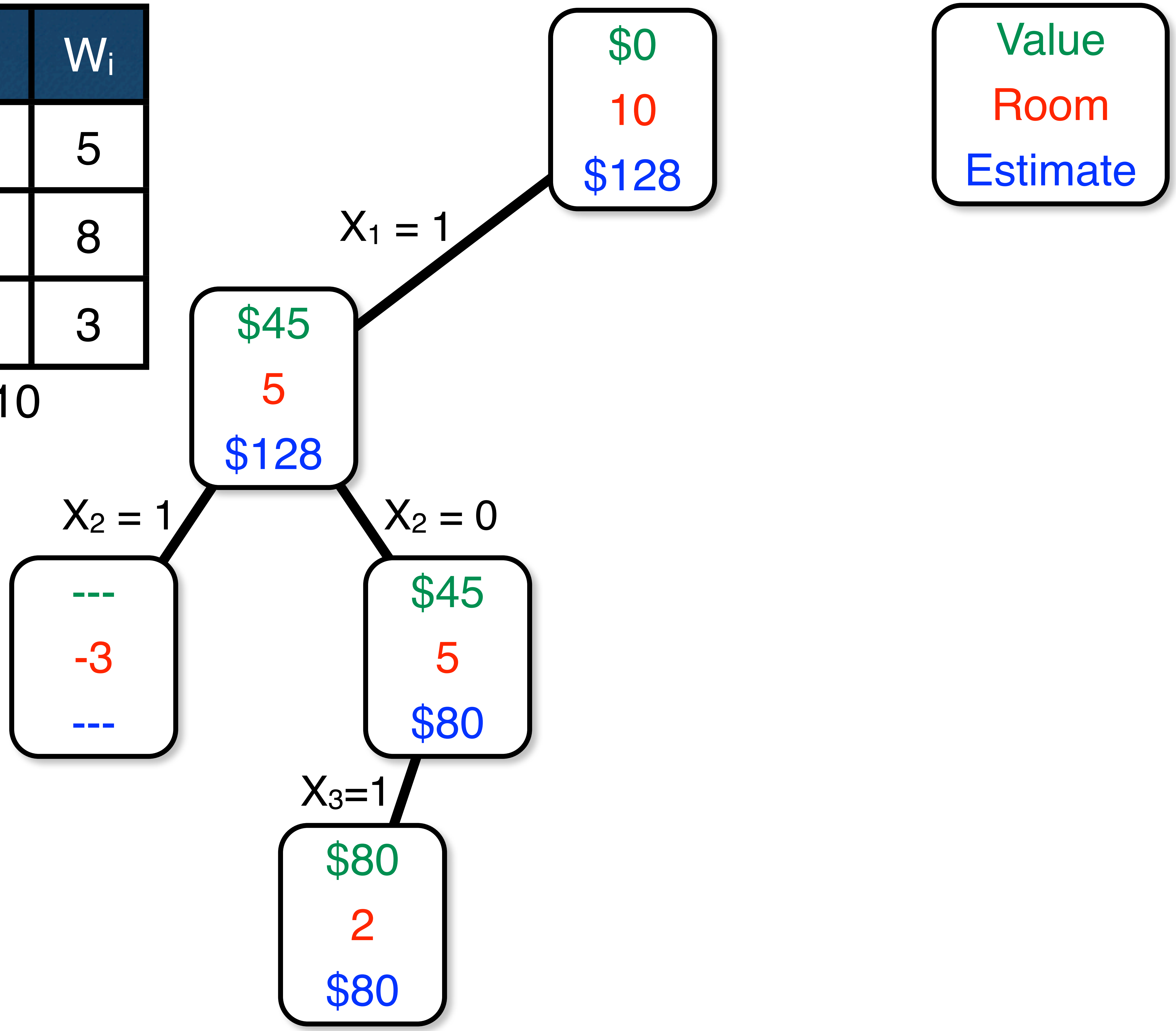


Value
Room
Estimate

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

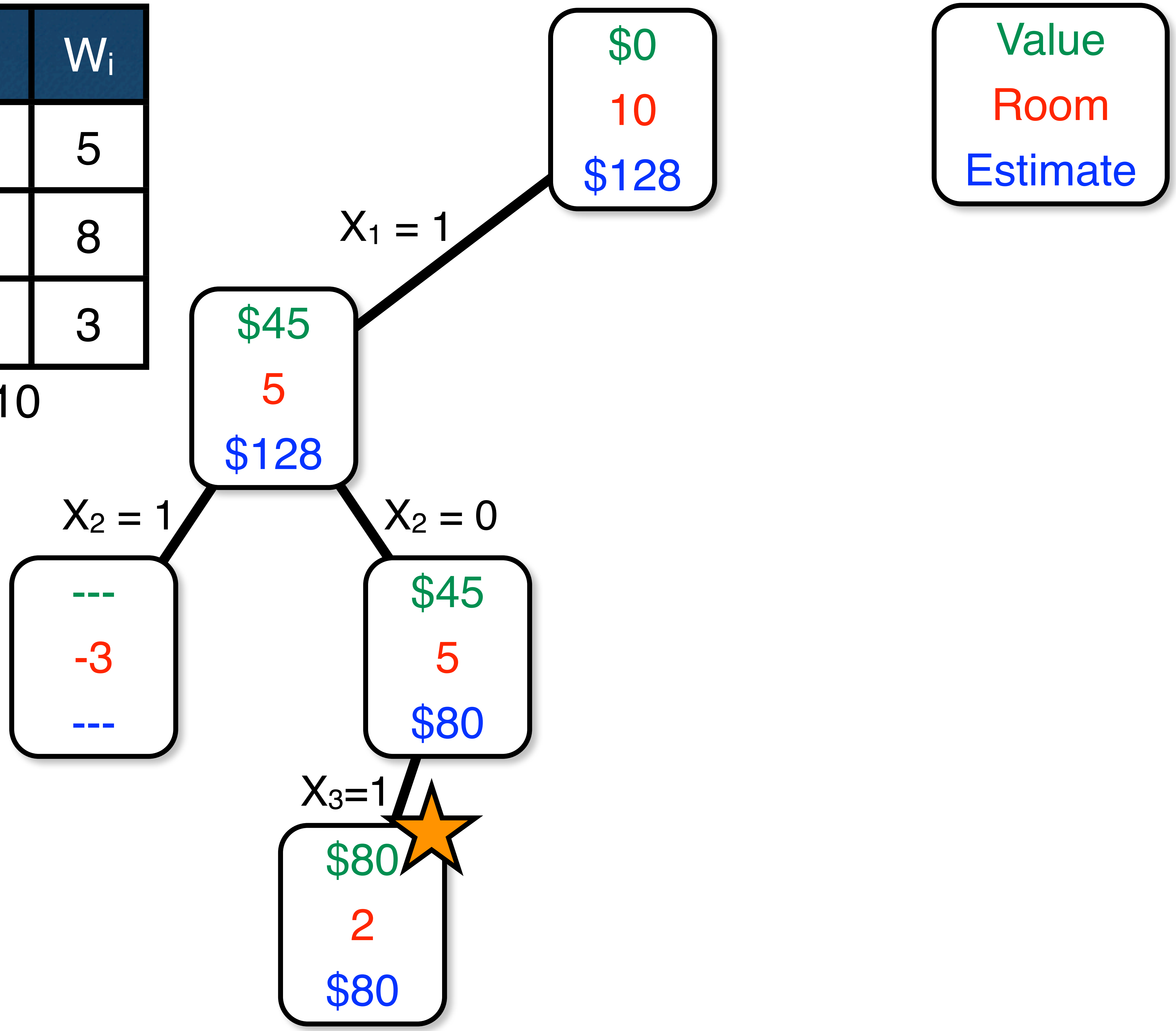
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

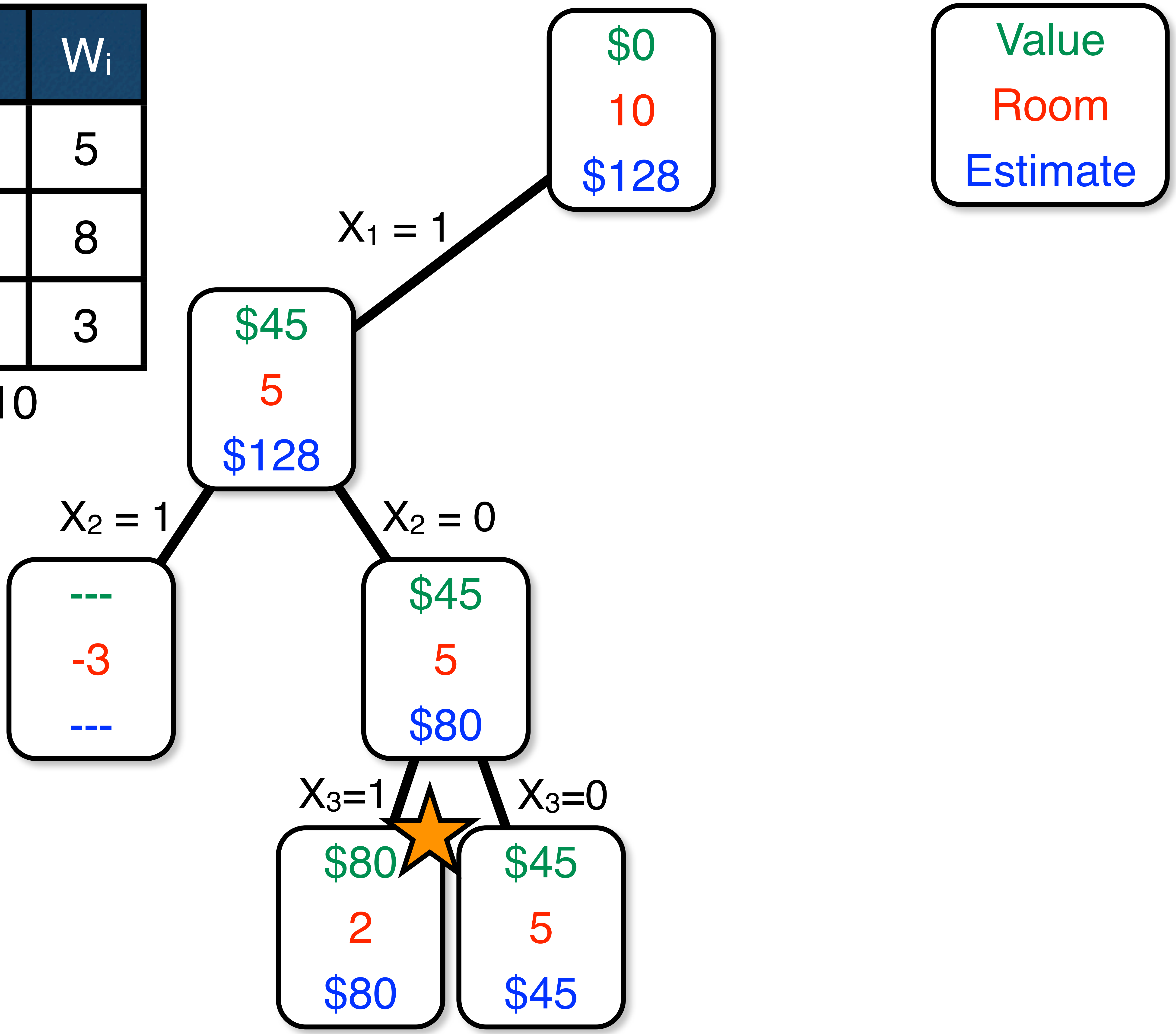
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

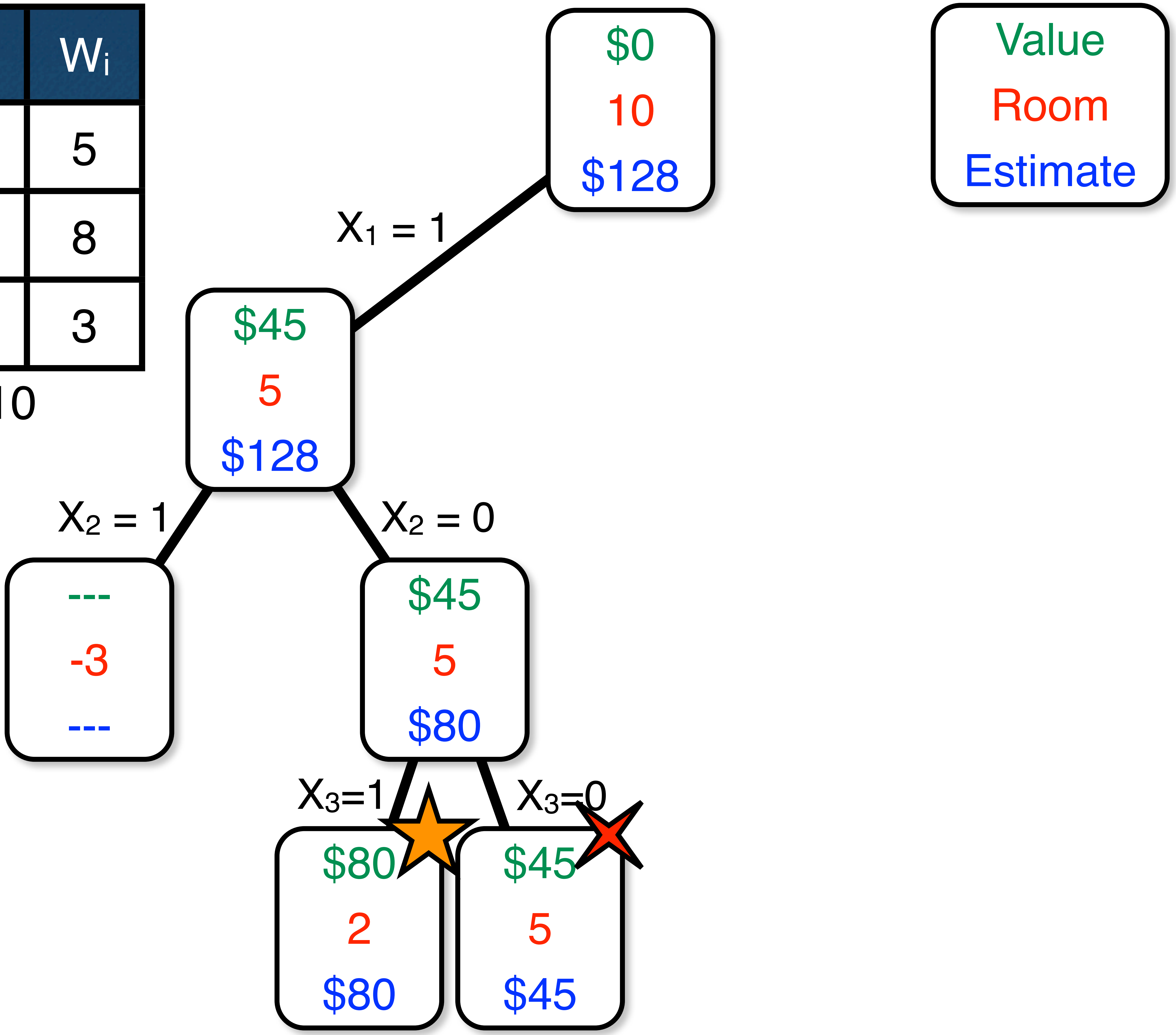
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

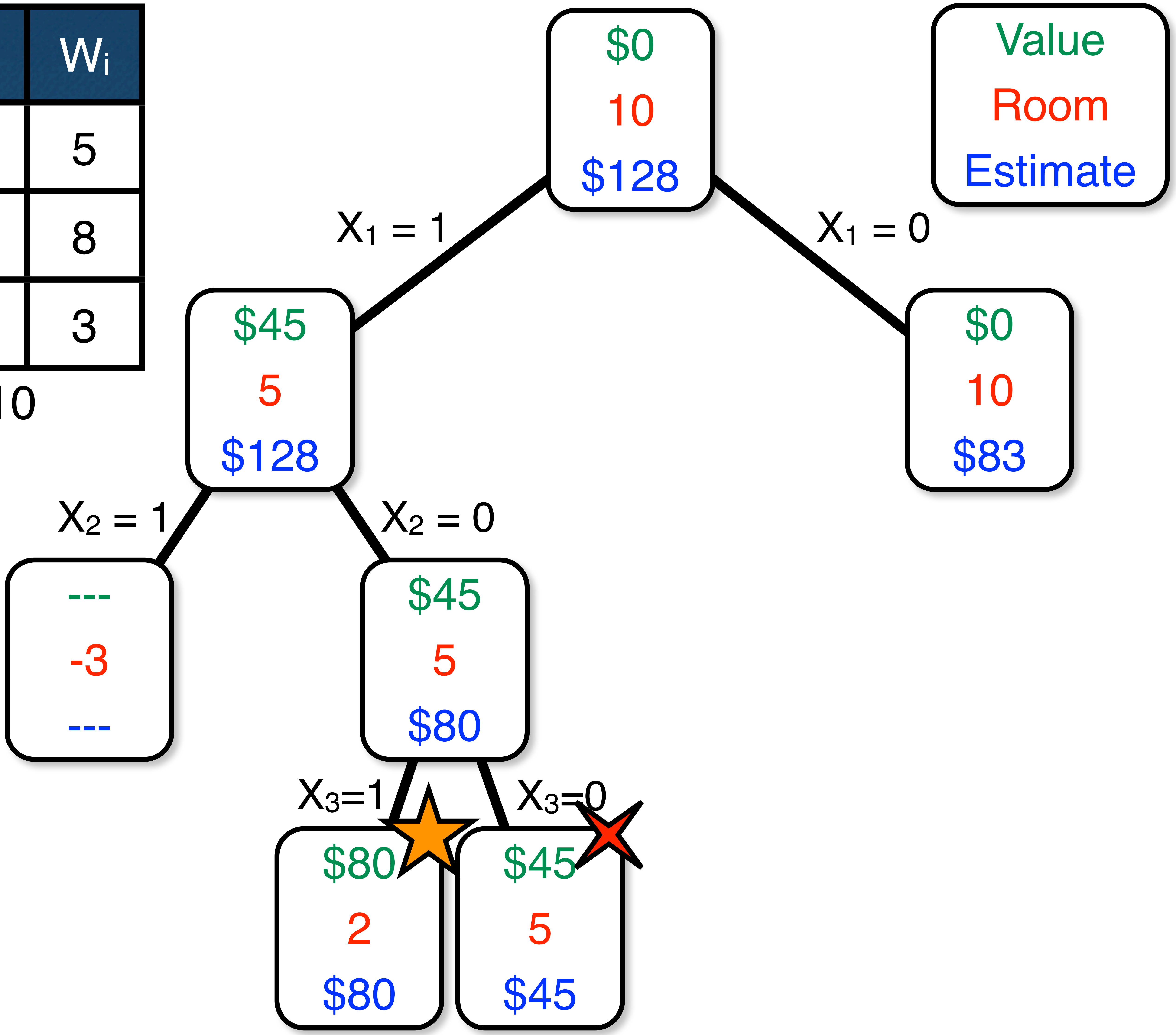
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

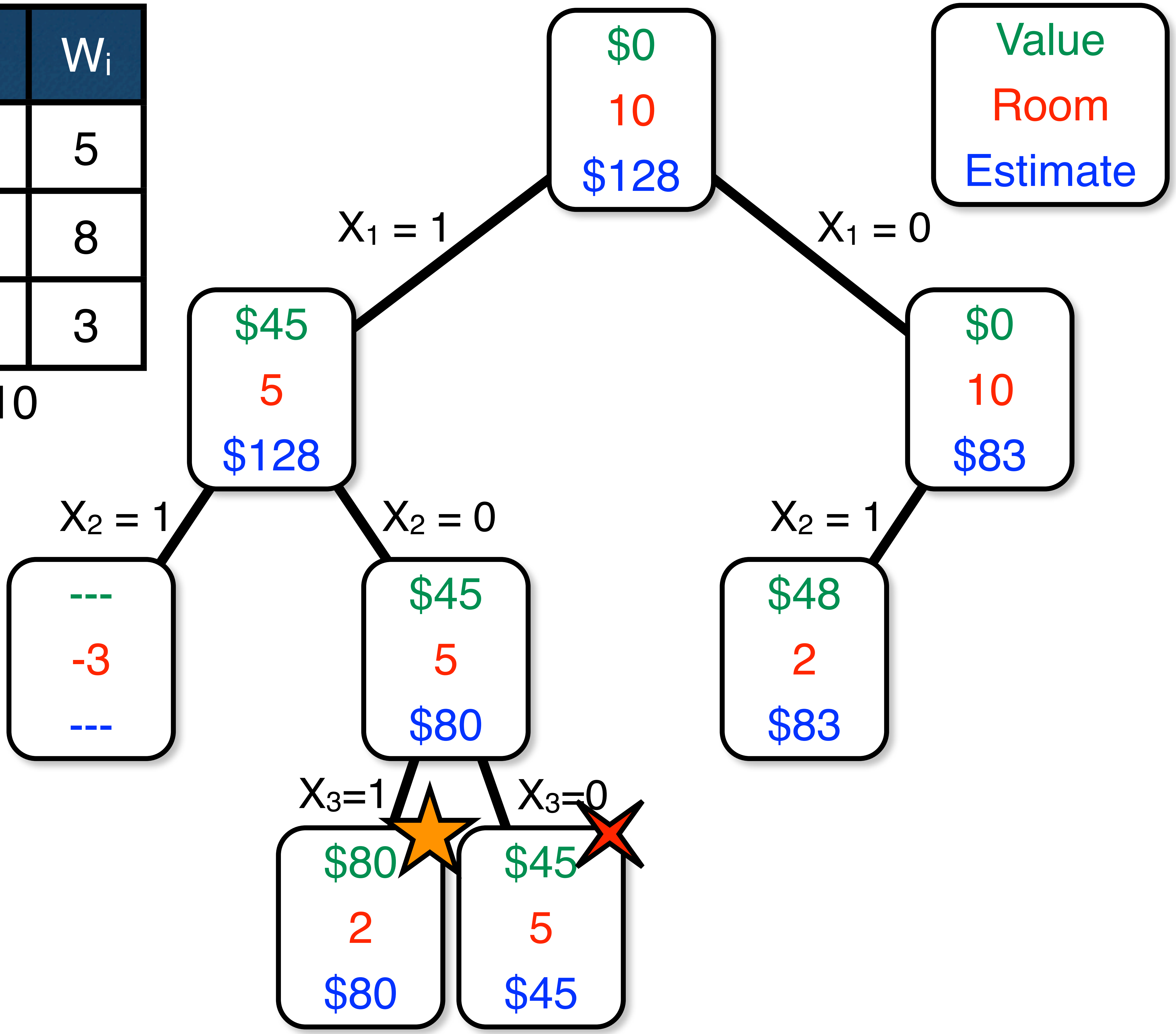
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

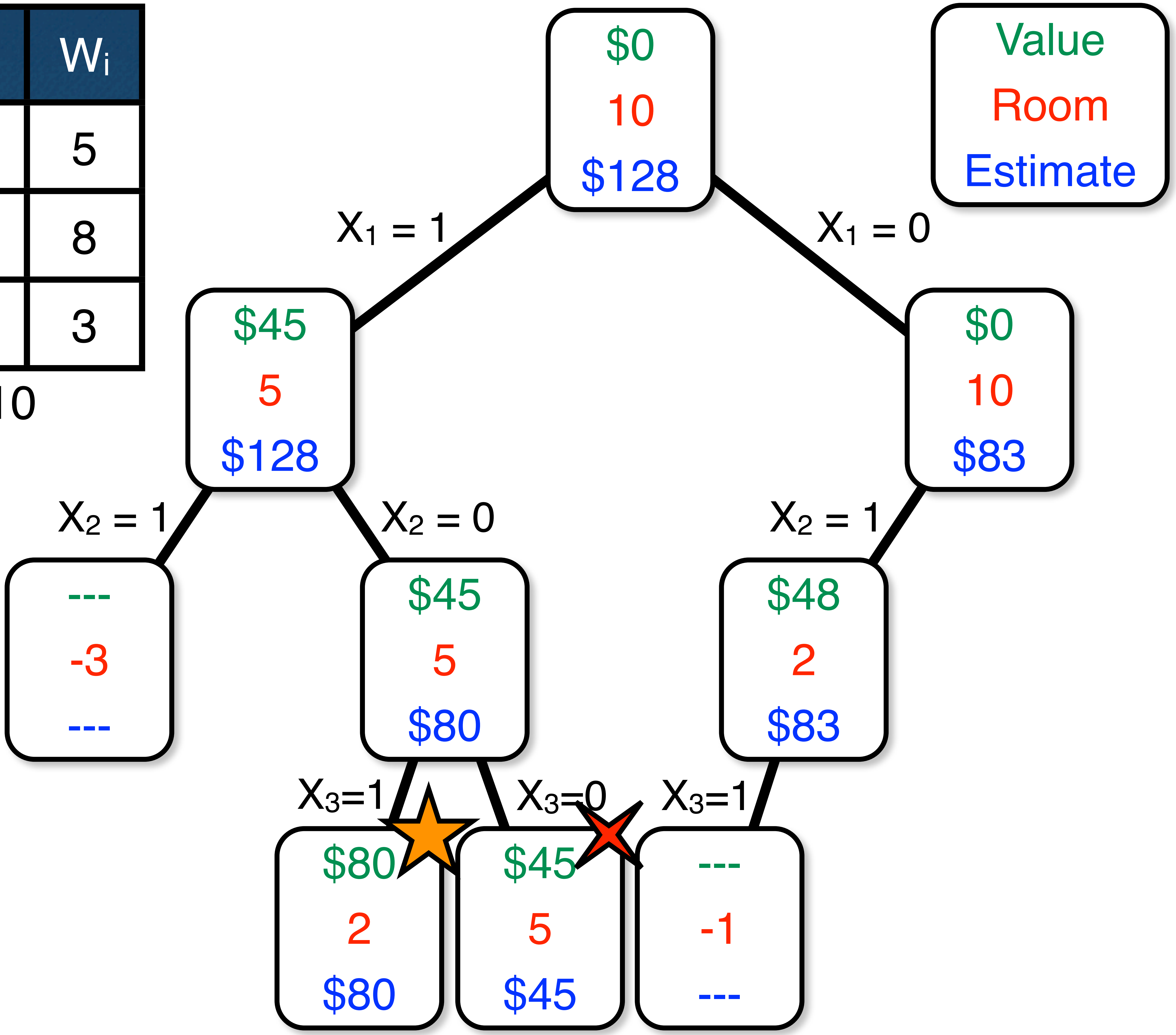
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

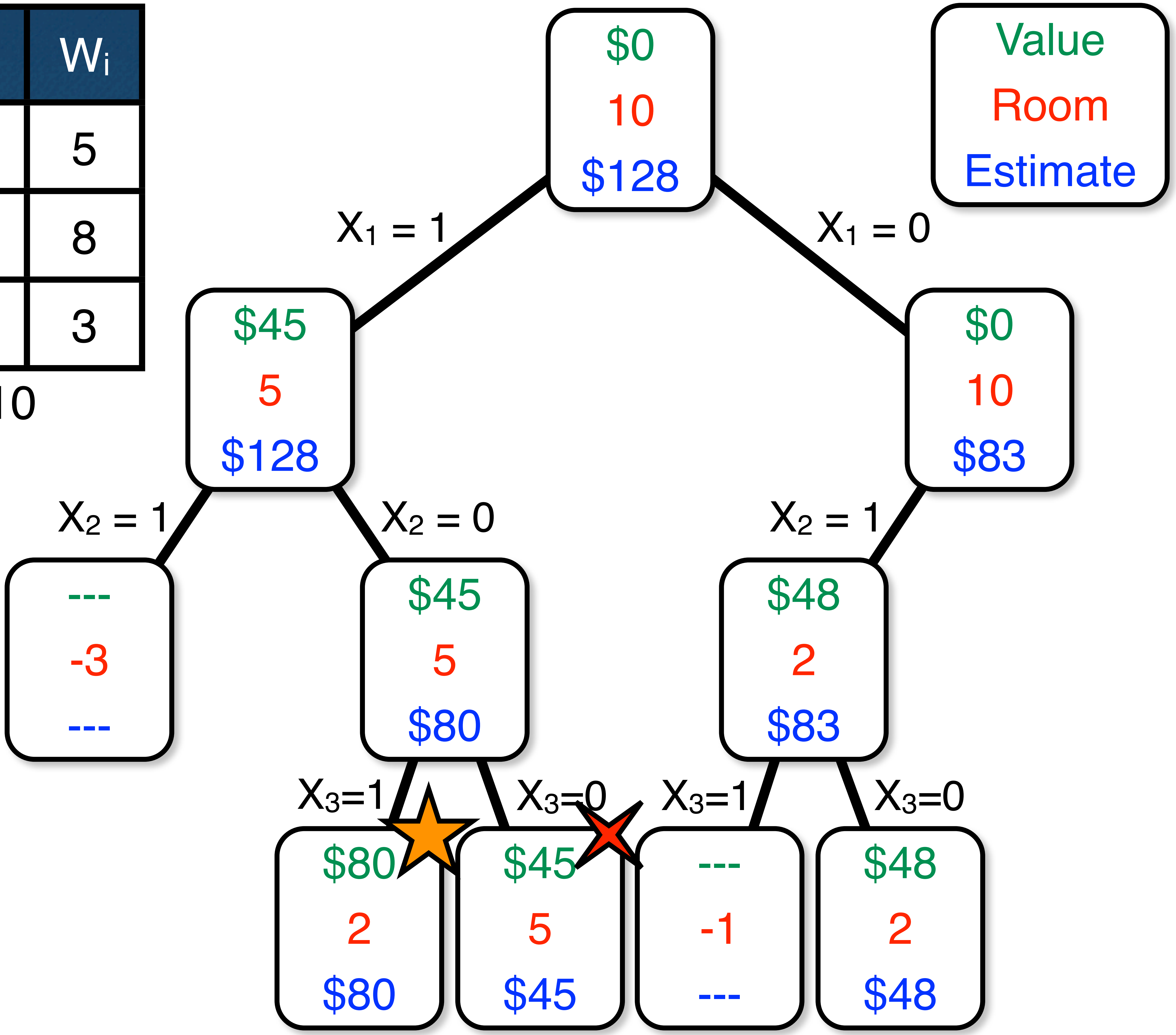
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

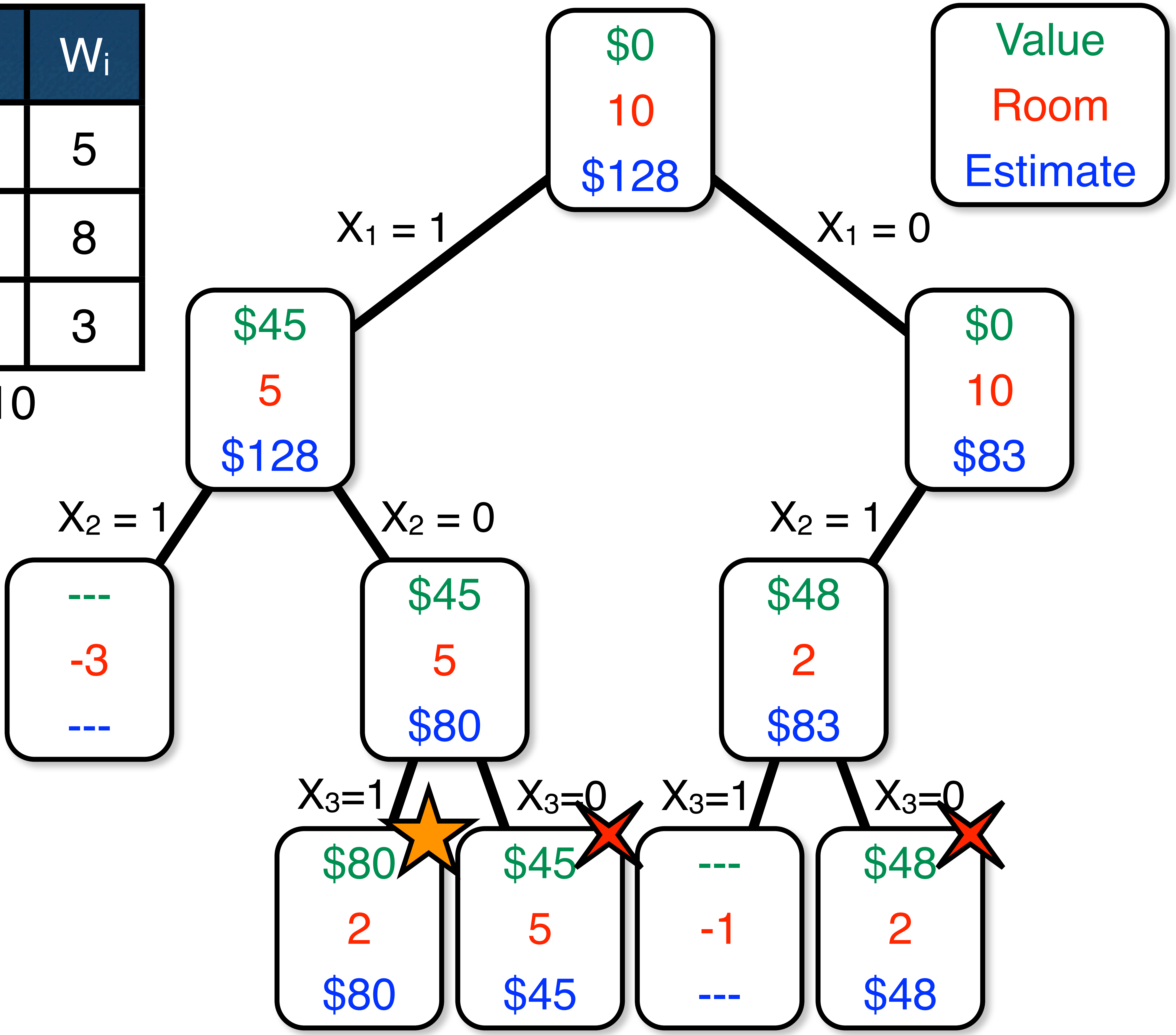
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

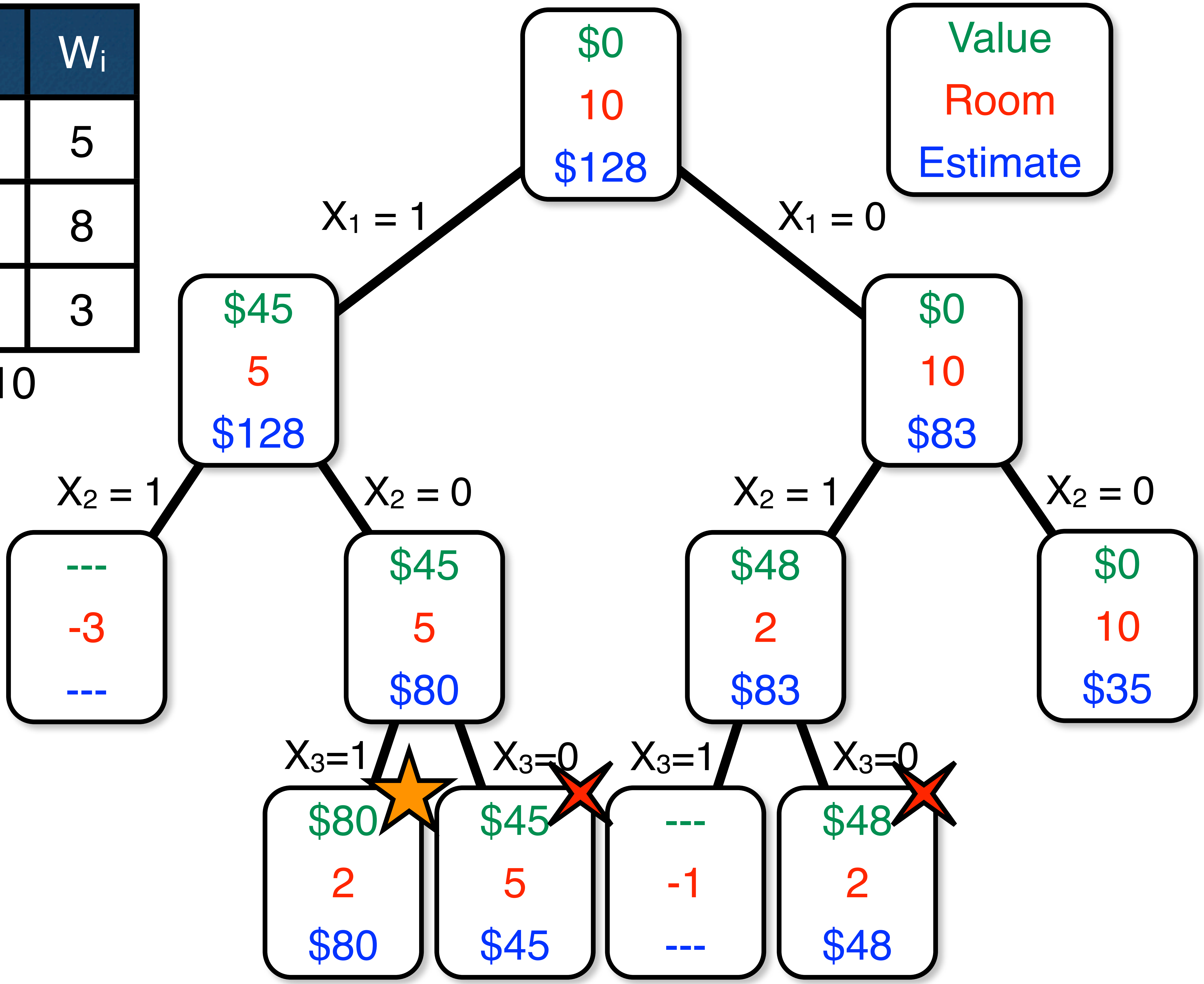
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

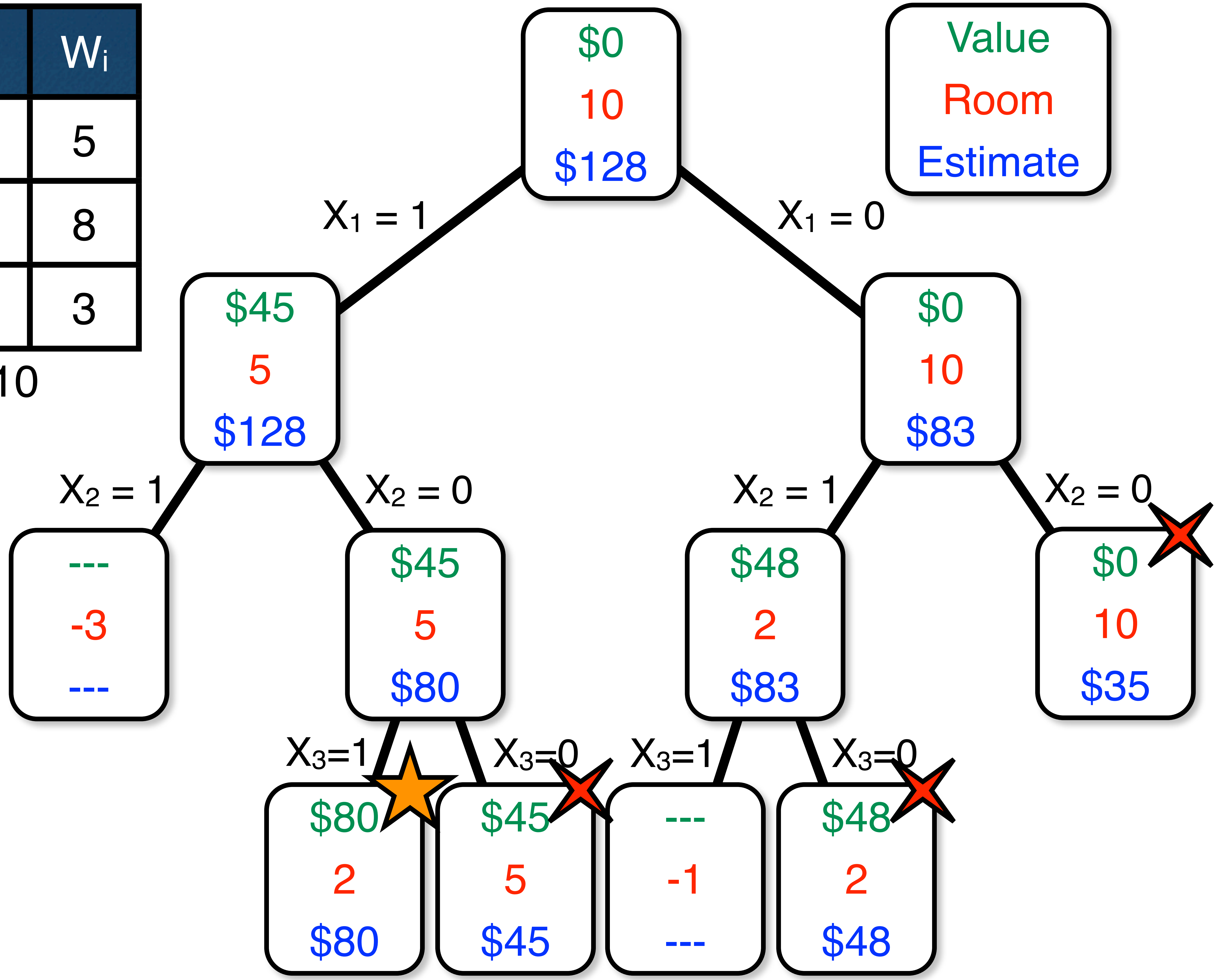
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10



A Knapsack Model

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & x_i \in \{0, 1\} \quad (i \in 1..3) \end{array}$$

A Knapsack Model

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & x_i \in \{0, 1\} \quad (i \in 1..3) \end{array}$$

- Can we relax something else?

A Knapsack Model

- What if the items are bars of Belgian chocolate?

A Knapsack Model

- What if the items are bars of Belgian chocolate?
  - In that case, we could actually take a fraction of the bar!

A Knapsack Model

- What if the items are bars of Belgian chocolate?
  - In that case, we could actually take a fraction of the bar!

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & \boxed{0 \leq x_i \leq 1} \quad (i \in 1..3) \end{array}$$

A Knapsack Model

- What if the items are bars of Belgian chocolate?
  - In that case, we could actually take a fraction of the bar!

$$\begin{array}{ll} \text{maximize} & 45x_1 + 48x_2 + 35x_3 \\ \text{subject to} & \\ & 5x_1 + 8x_2 + 3x_3 \leq 10 \\ & \boxed{0 \leq x_i \leq 1} \quad (i \in 1..3) \end{array}$$

- This is called the linear relaxation
  - we will come back to this later in the class
  - we relax the integrality requirement

A Knapsack Model

- Can we solve a knapsack when we can take parts of the items?

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
 & 5x_1 + 8x_2 + 3x_3 \leq 10 \\
 & 0 \leq x_i \leq 1 \quad (i \in 1..3)
\end{array}
$$

A Knapsack Model

- Can we solve a knapsack when we can take parts of the items?
  - order the items by decreasing value of V_i/W_i

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
& 5x_1 + 8x_2 + 3x_3 \leq 10 \\
& 0 \leq x_i \leq 1 \quad (i \in 1..3)
\end{array}
$$

A Knapsack Model

- Can we solve a knapsack when we can take parts of the items?
  - order the items by decreasing value of V_i/W_i
  - "most value per kilo"

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
& 5x_1 + 8x_2 + 3x_3 \leq 10 \\
& 0 \leq x_i \leq 1 \quad (i \in 1..3)
\end{array}
$$

A Knapsack Model

- How to solve the relaxation now?
  - select the items while the capacity is not exhausted
  - select a fraction of the last item

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
& 5x_1 + 8x_2 + 3x_3 \leq 10 \\
& 0 \leq x_i \leq 1 \quad (i \in 1..3)
\end{array}
$$

A Knapsack Model

- How to solve the relaxation now?
  - select the items while the capacity is not exhausted
  - select a fraction of the last item

$$
\begin{array}{ll}
\text{maximize} & 45x_1 + 48x_2 + 35x_3 \\
\text{subject to} & \\
& 5x_1 + 8x_2 + 3x_3 \leq 10 \\
& 0 \leq x_i \leq 1 \quad (i \in 1..3)
\end{array}
$$

- In this example,
  - $V_1/W_1 = 9$, $V_2/W_2 = 6$, $V_3/W_3 = 11.7$
  - select items 3 and 1
  - select 1/4 of item 2
  - estimation: 92

A Knapsack Model

- Why is correct?

A Knapsack Model

- Why is correct?

$$\text{let } x_i = \frac{y_i}{v_i}$$

A Knapsack Model

- Why is correct?

$$\text{let } x_i = \frac{y_i}{v_i}$$

$$\begin{array}{ll} \text{maximize} & \sum_{i \in 1..j} y_i \\ \text{subject to} & \\ & \sum_{i \in 1..j} \frac{w_i}{v_i} y_i \leq K \\ & 0 \leq y_i \leq 1 \quad (i \in 1..j) \end{array}$$

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

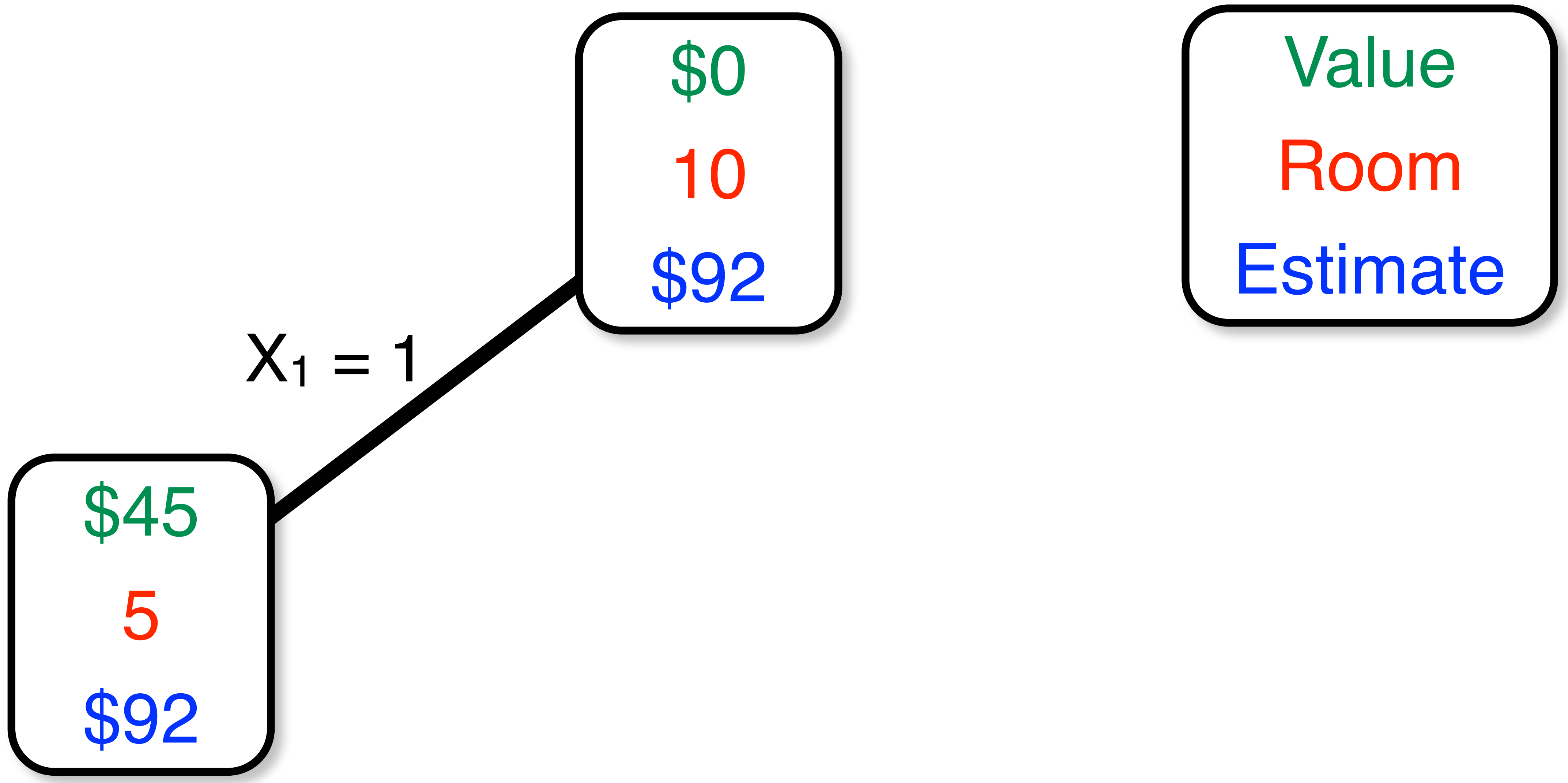
$0
10
$92

Value
Room
Estimate

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

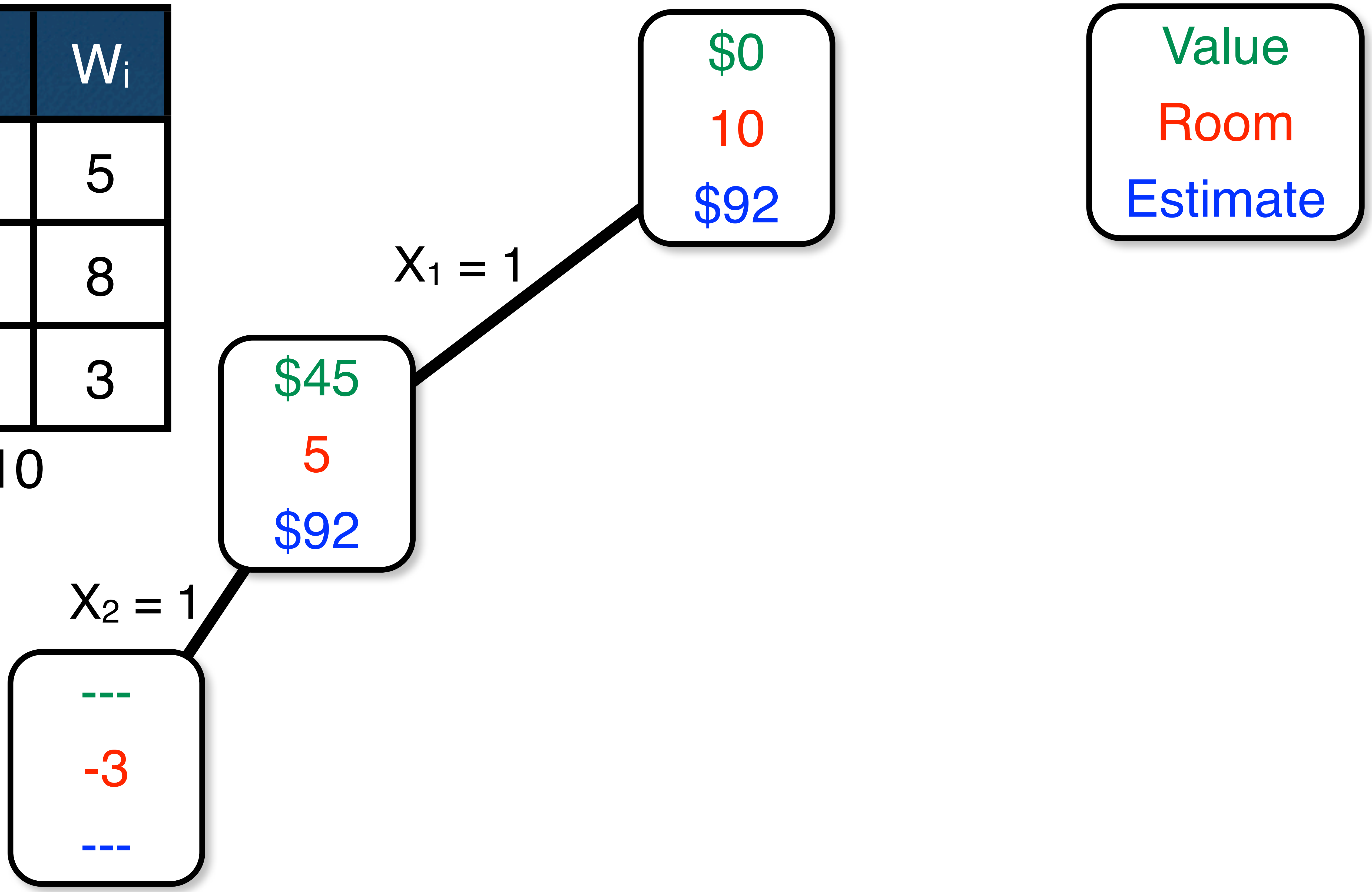
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

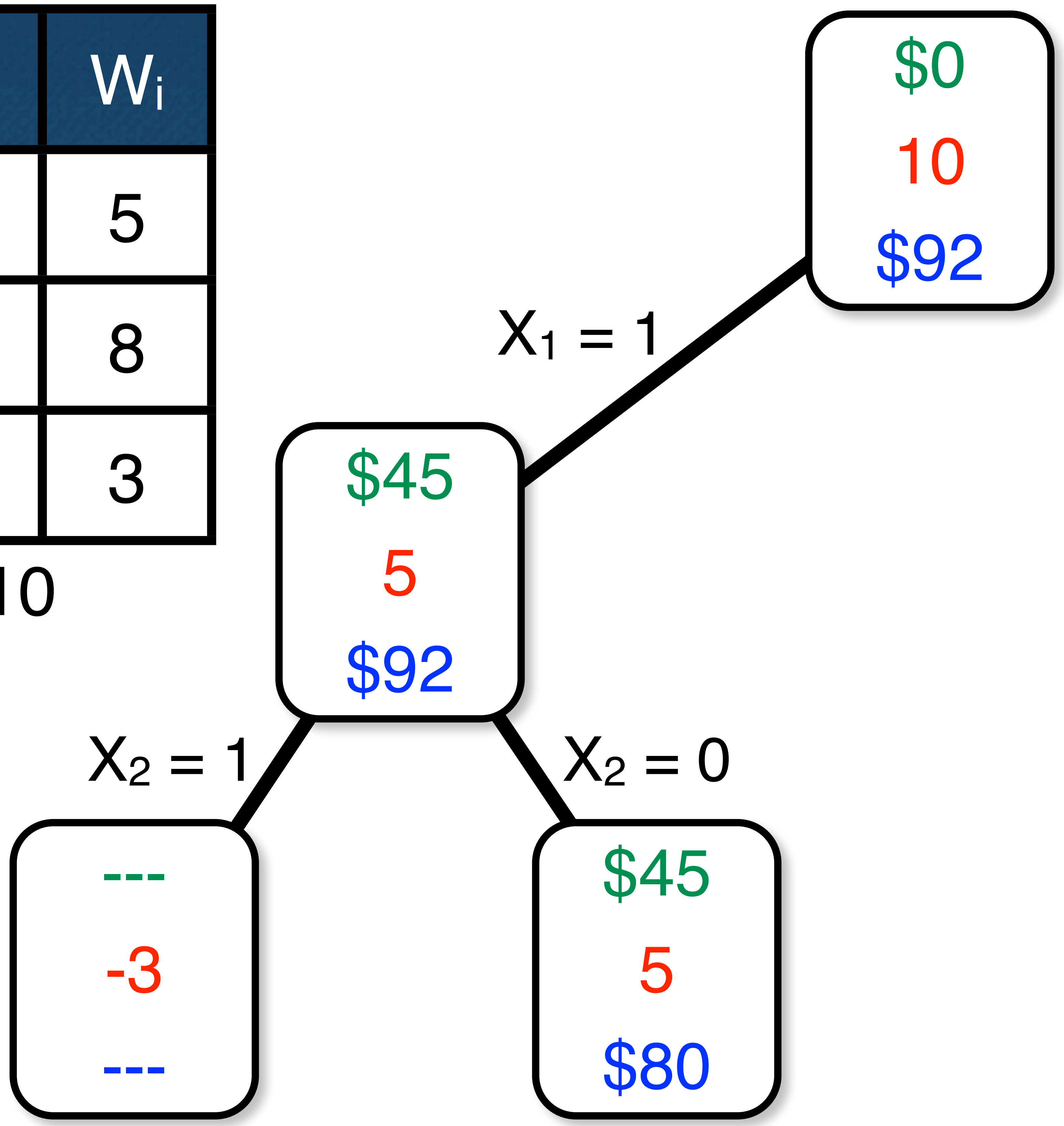
K = 10

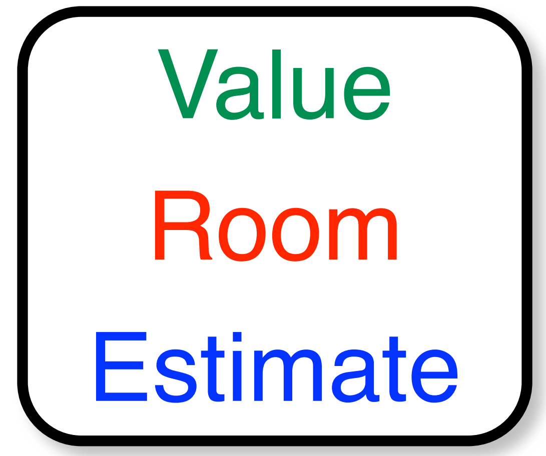


Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

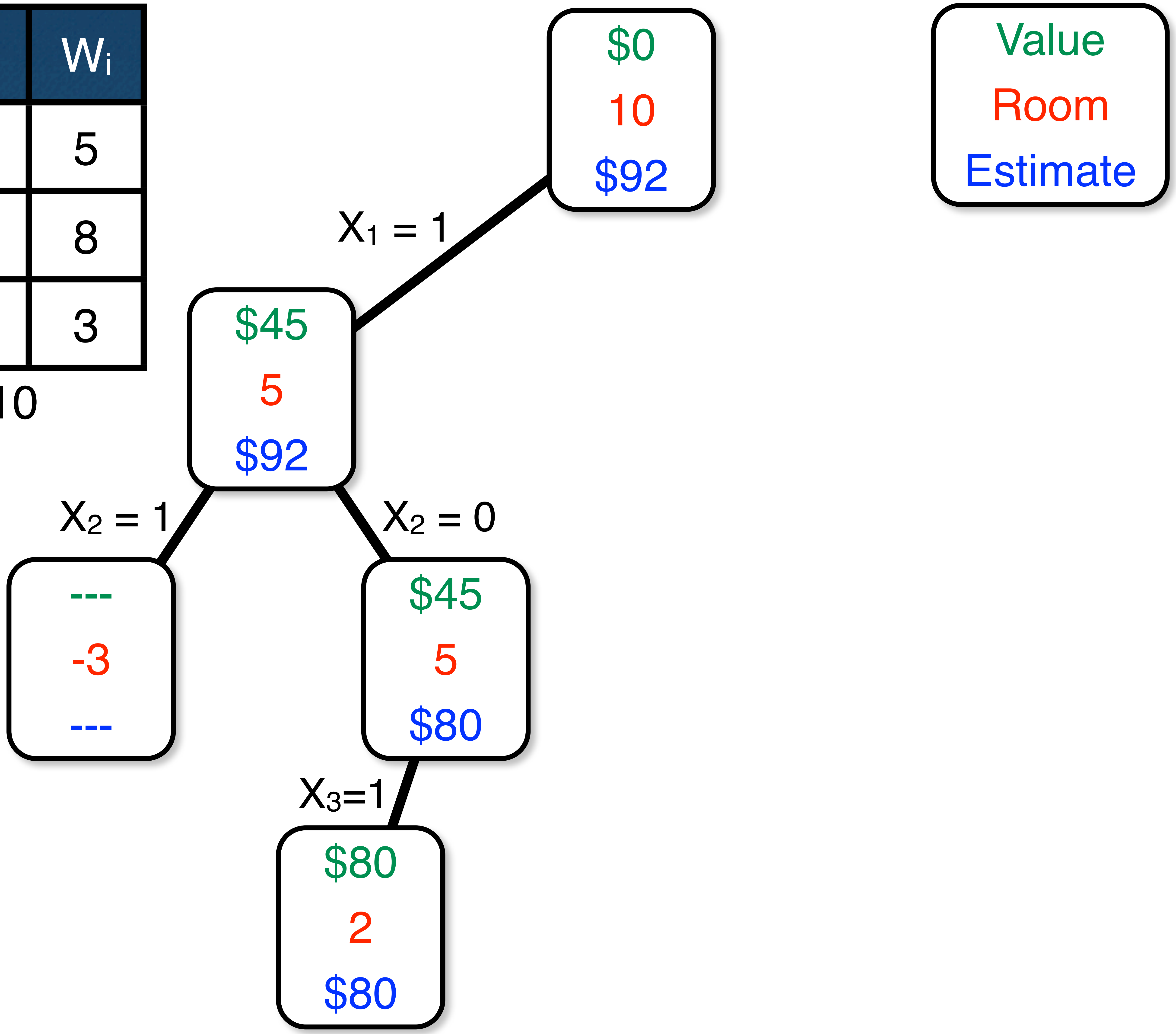




Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

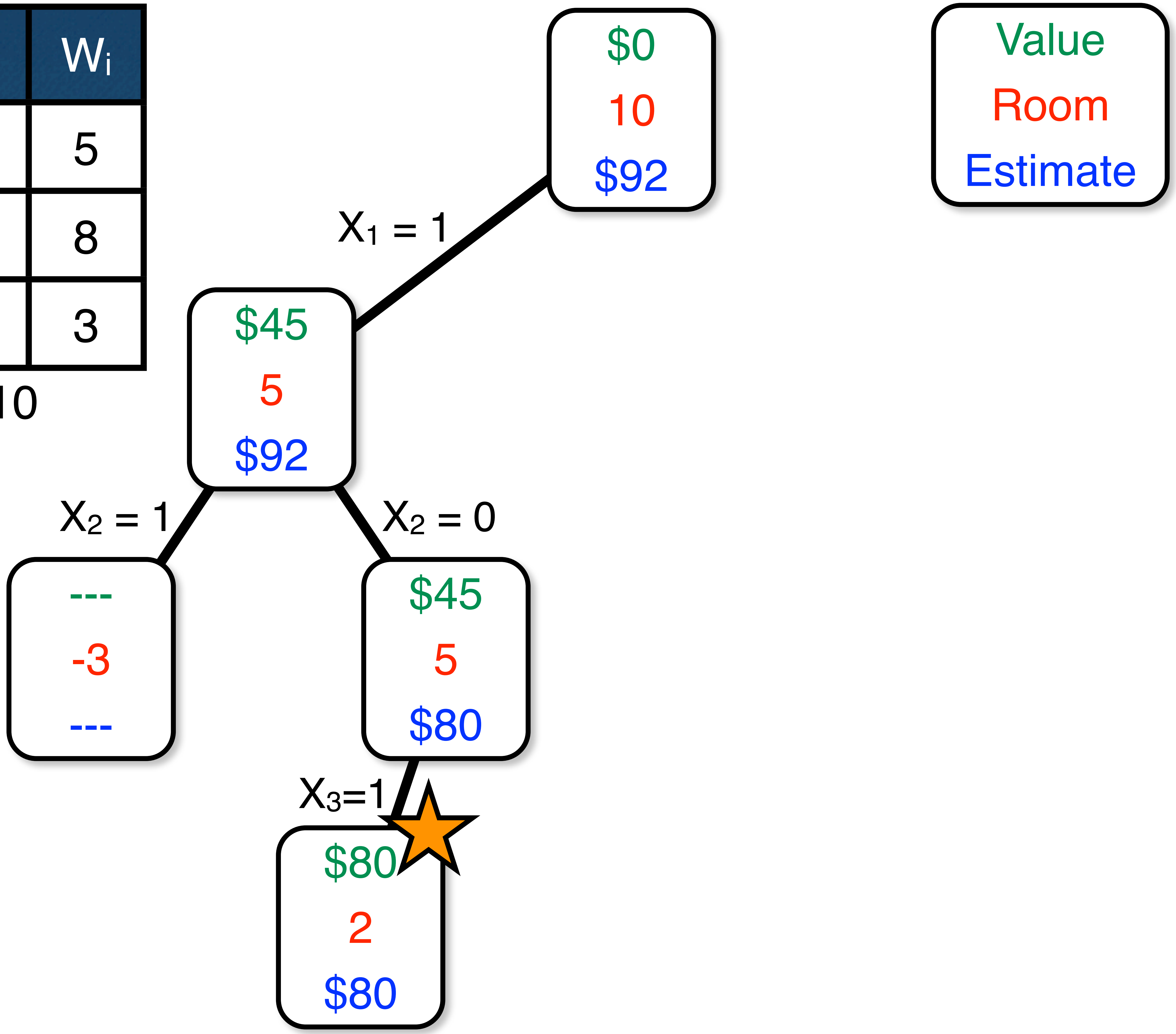
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

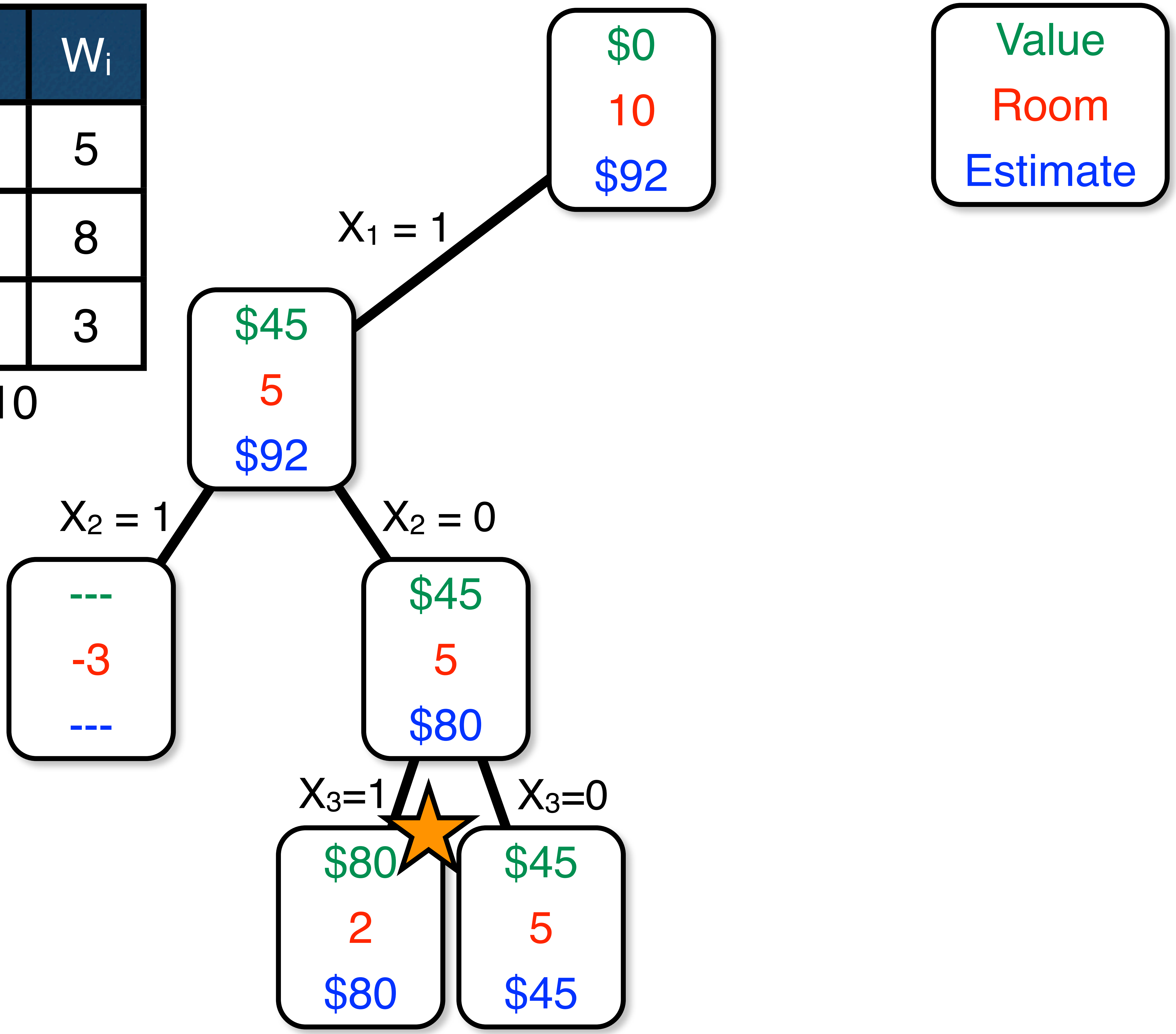
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

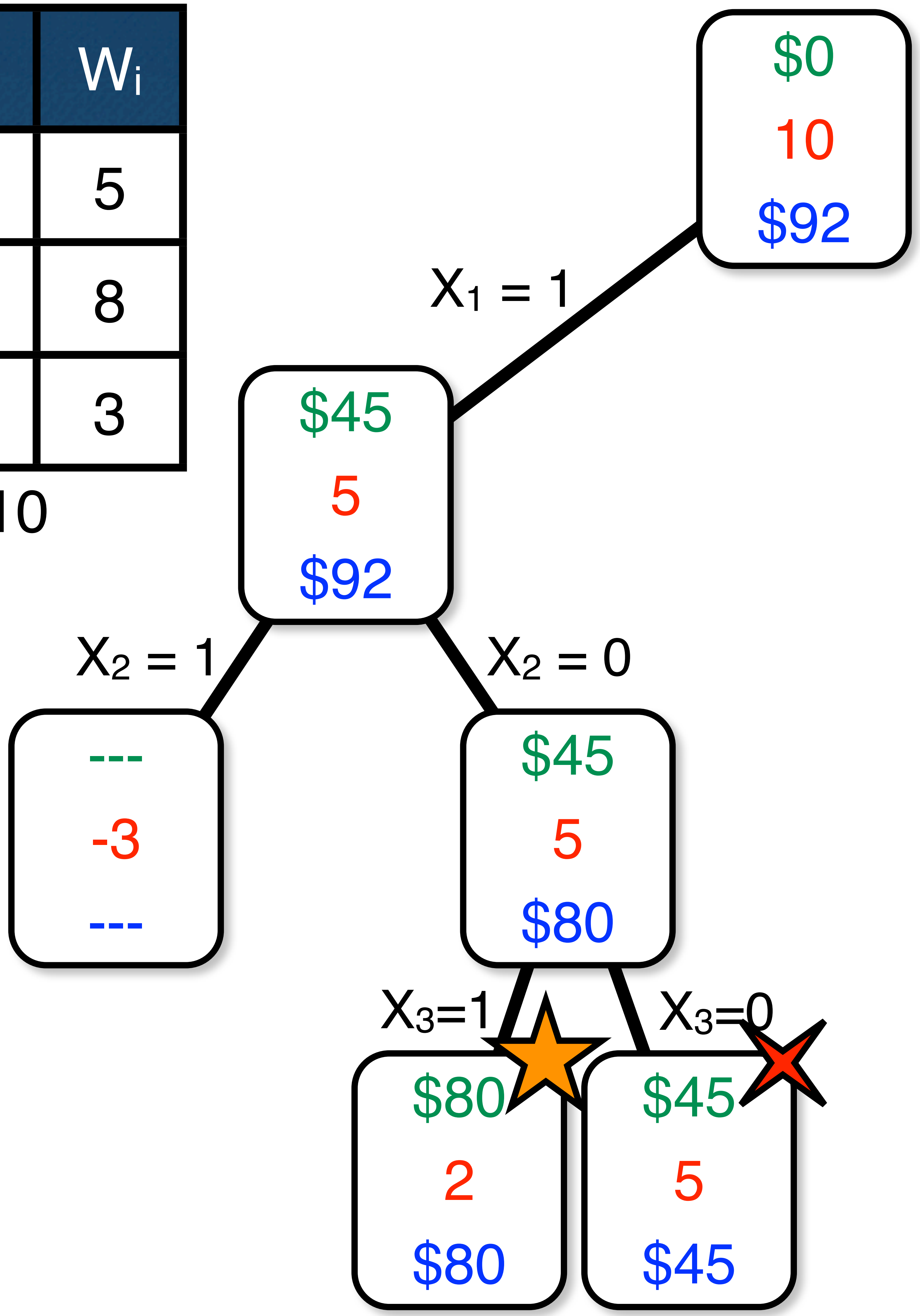


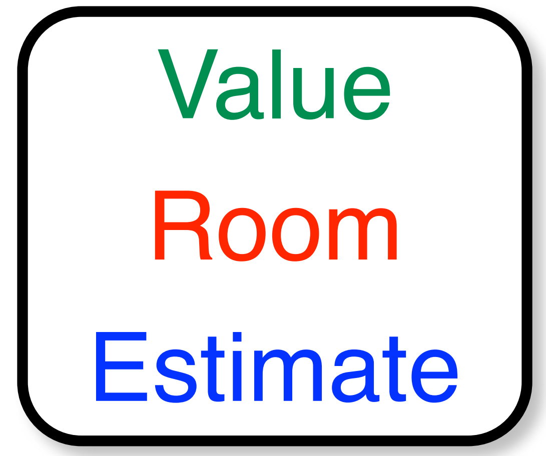
Thursday, 13 June 13

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

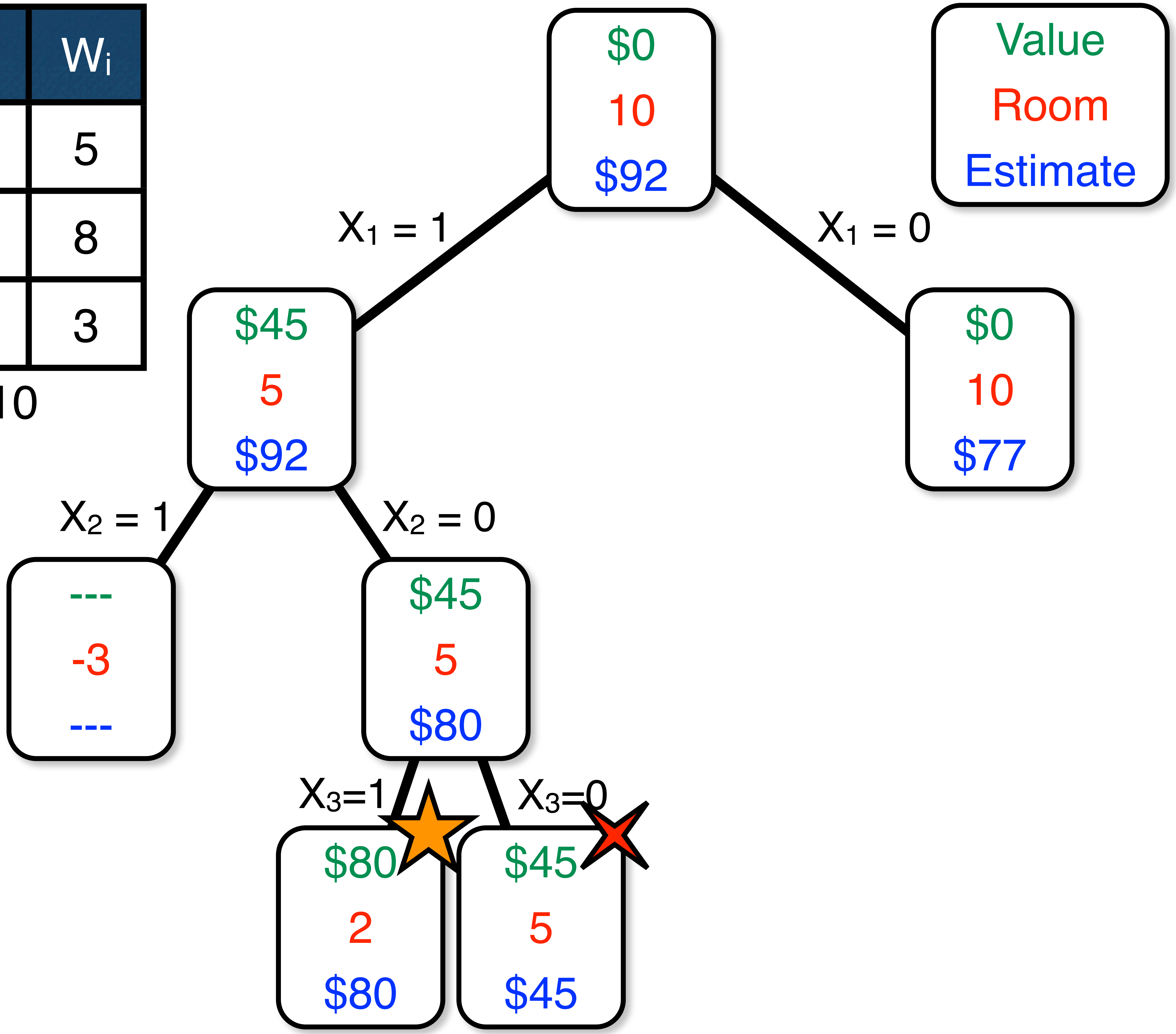




Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

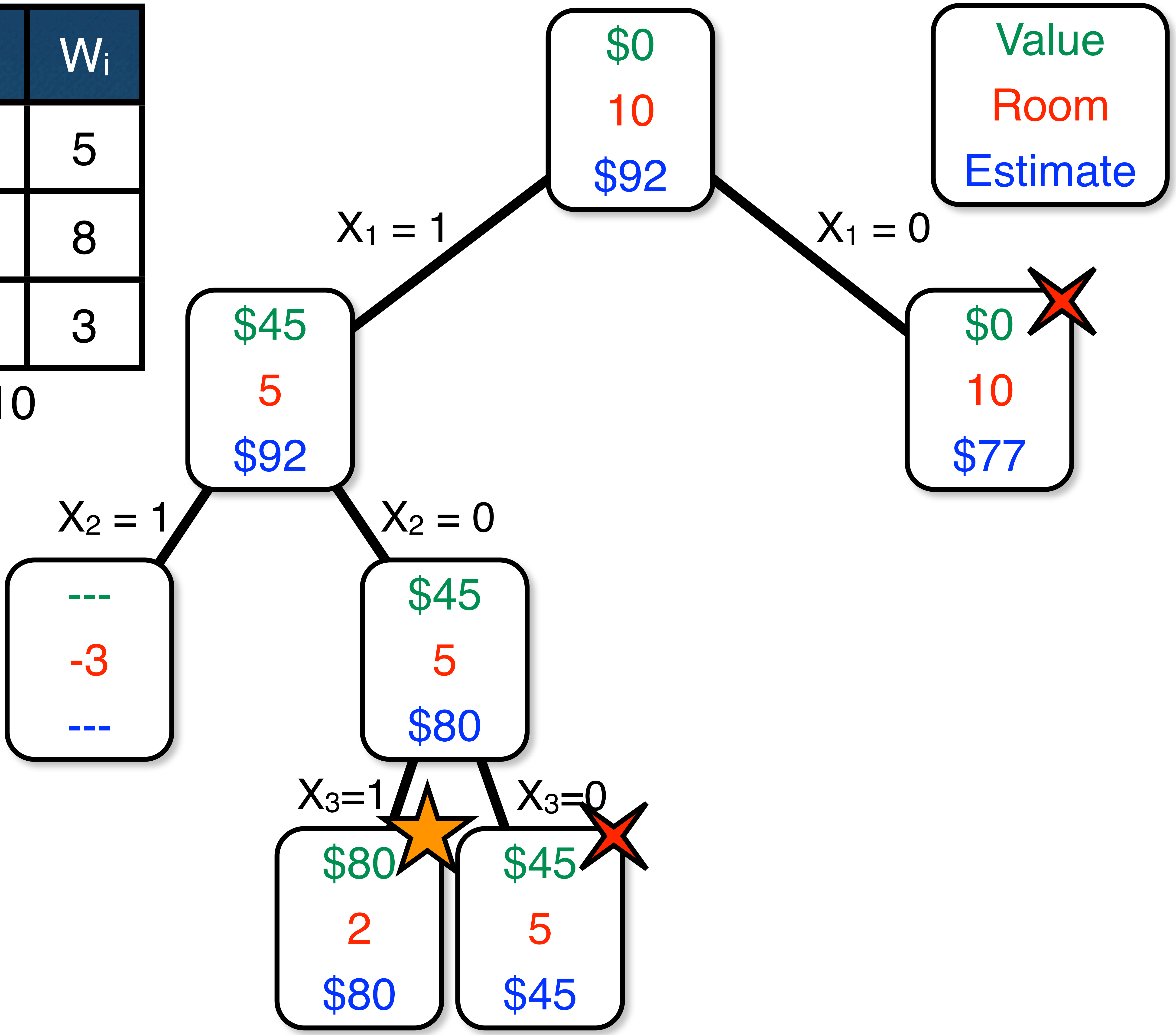
K = 10



Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10



Branch and Bound

- Search Strategies
  - depth-first
  - best-first
  - many others

Branch and Bound

- Search Strategies
  - depth-first
  - best-first
  - many others
- Depth-first
  - prunes when a node estimation is worse than the best found solution
  - memory efficient

Branch and Bound

- Search Strategies
  - depth-first
  - best-first
  - many others
- Depth-first
  - prunes when a node estimation is worse than the best found solution
  - memory efficient
- Best-First
  - select the node with the best estimation

Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10

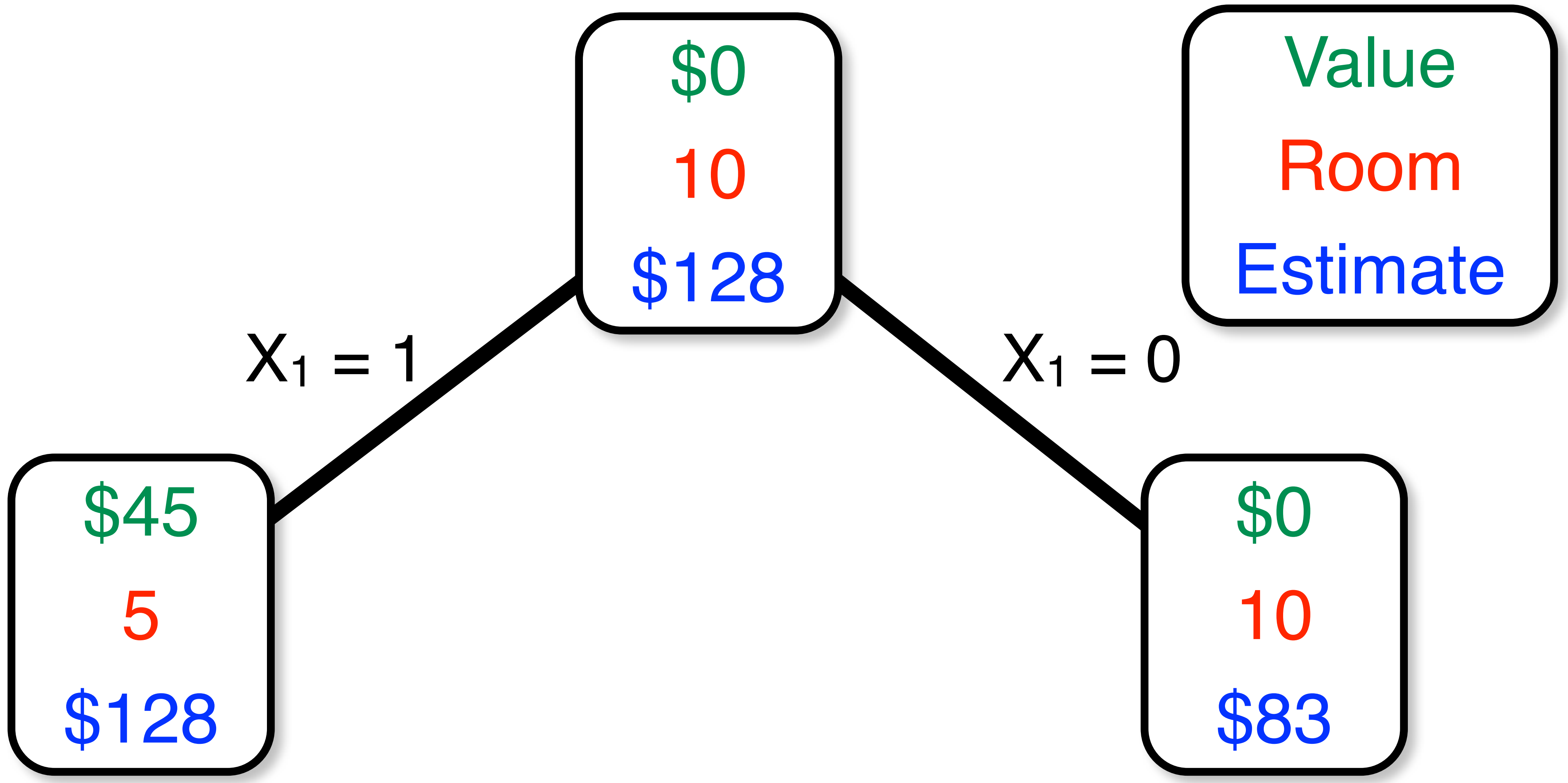
$0
10
$128

Value
Room
Estimate

Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

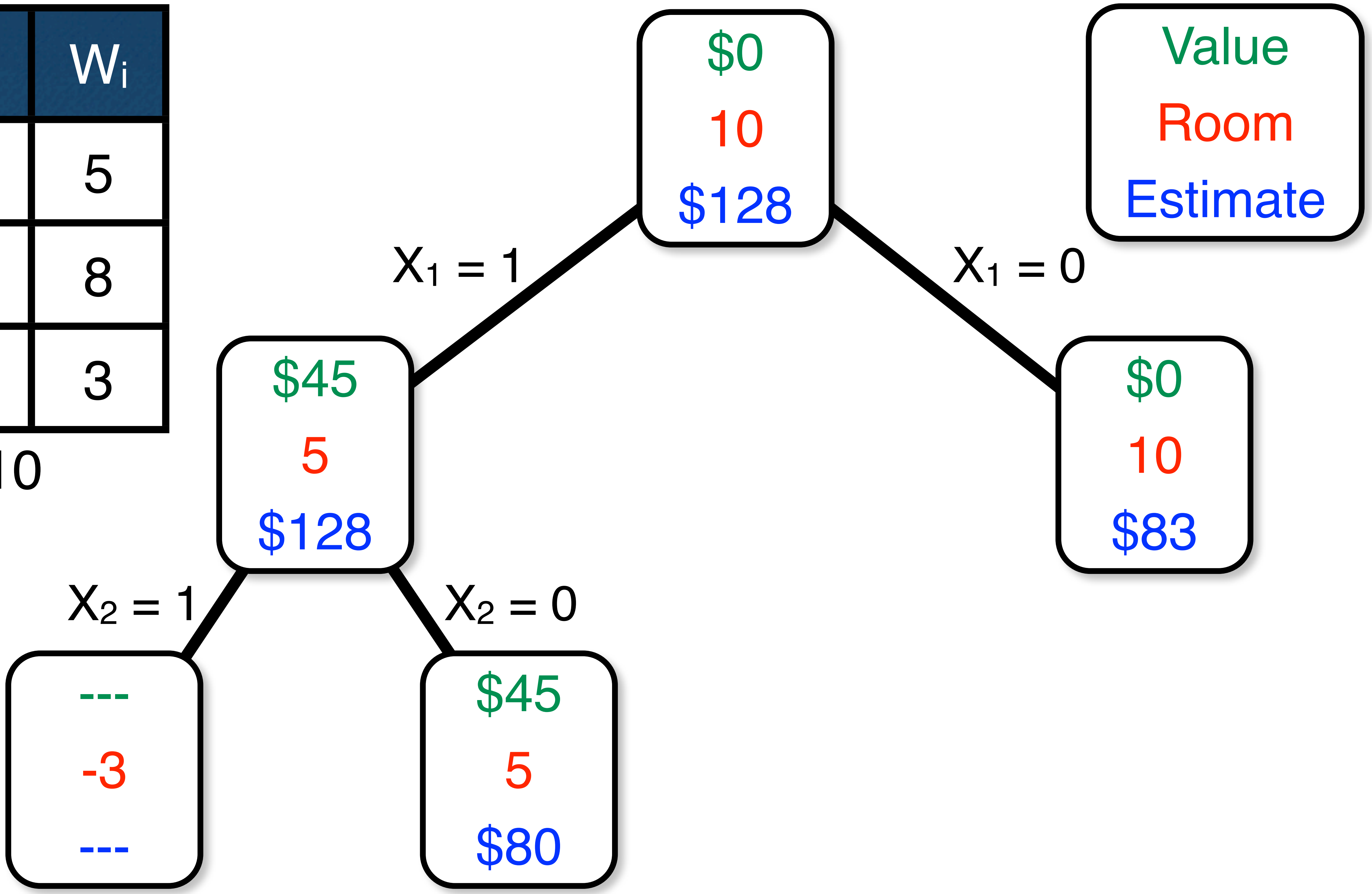
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

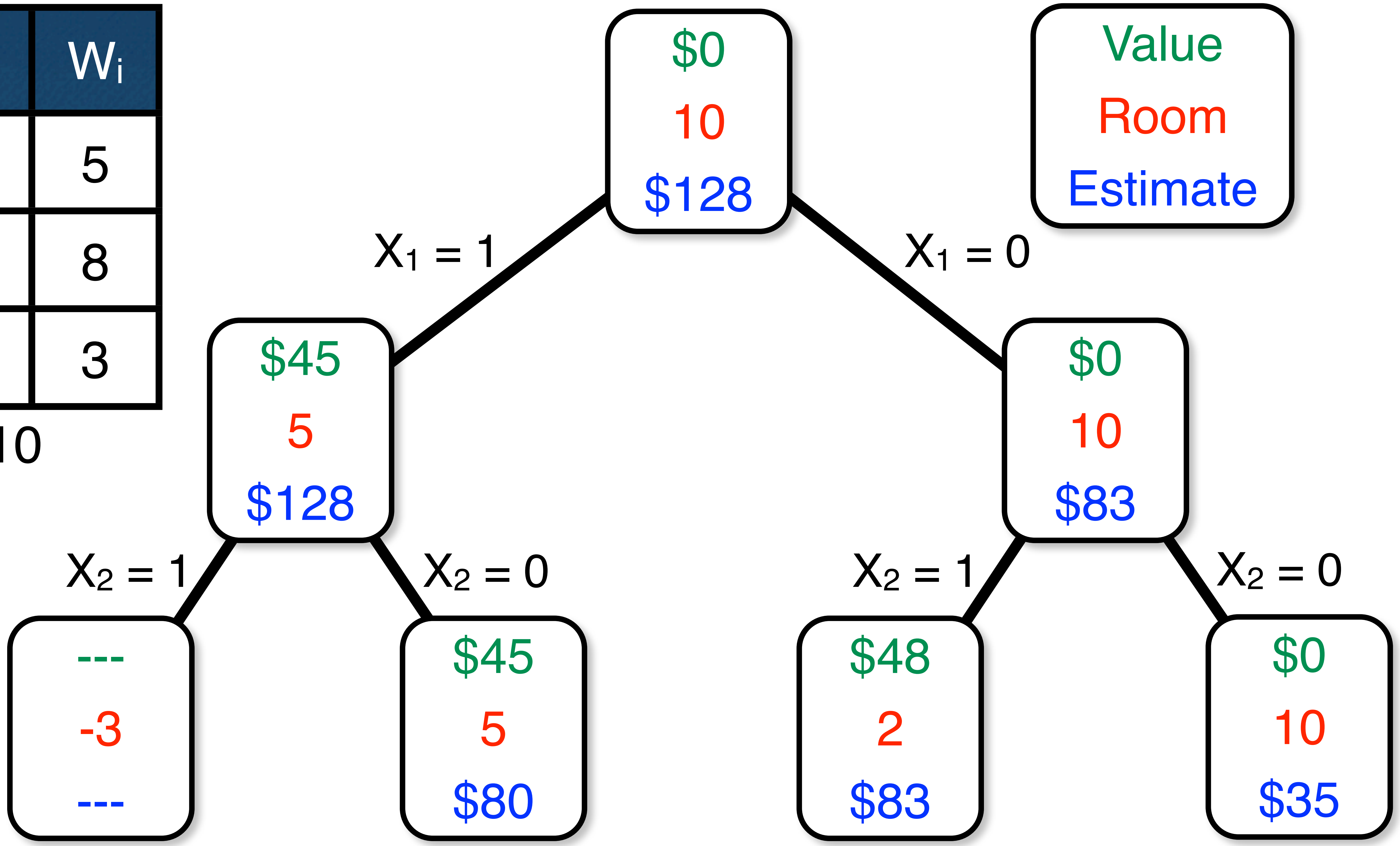
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

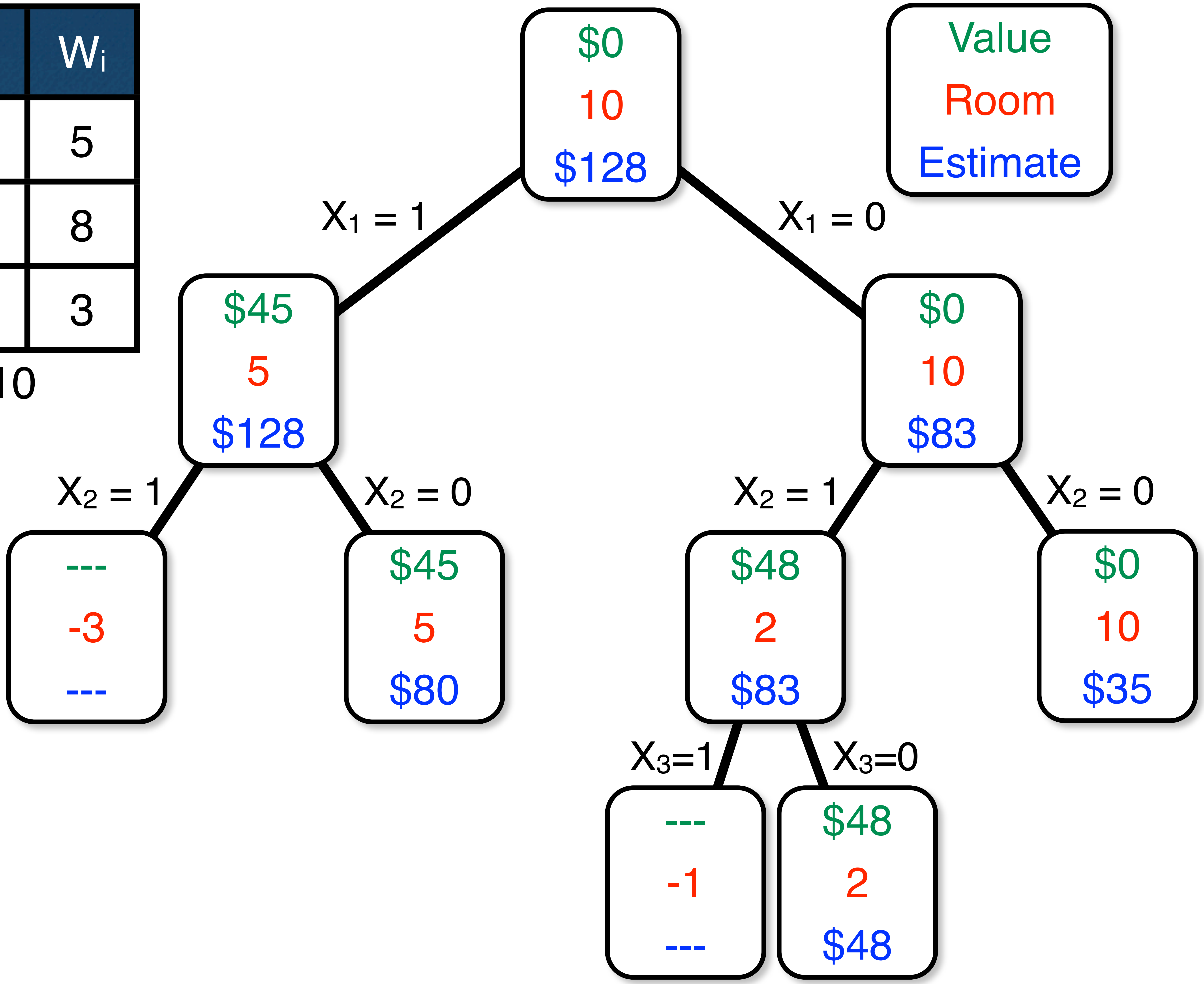
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

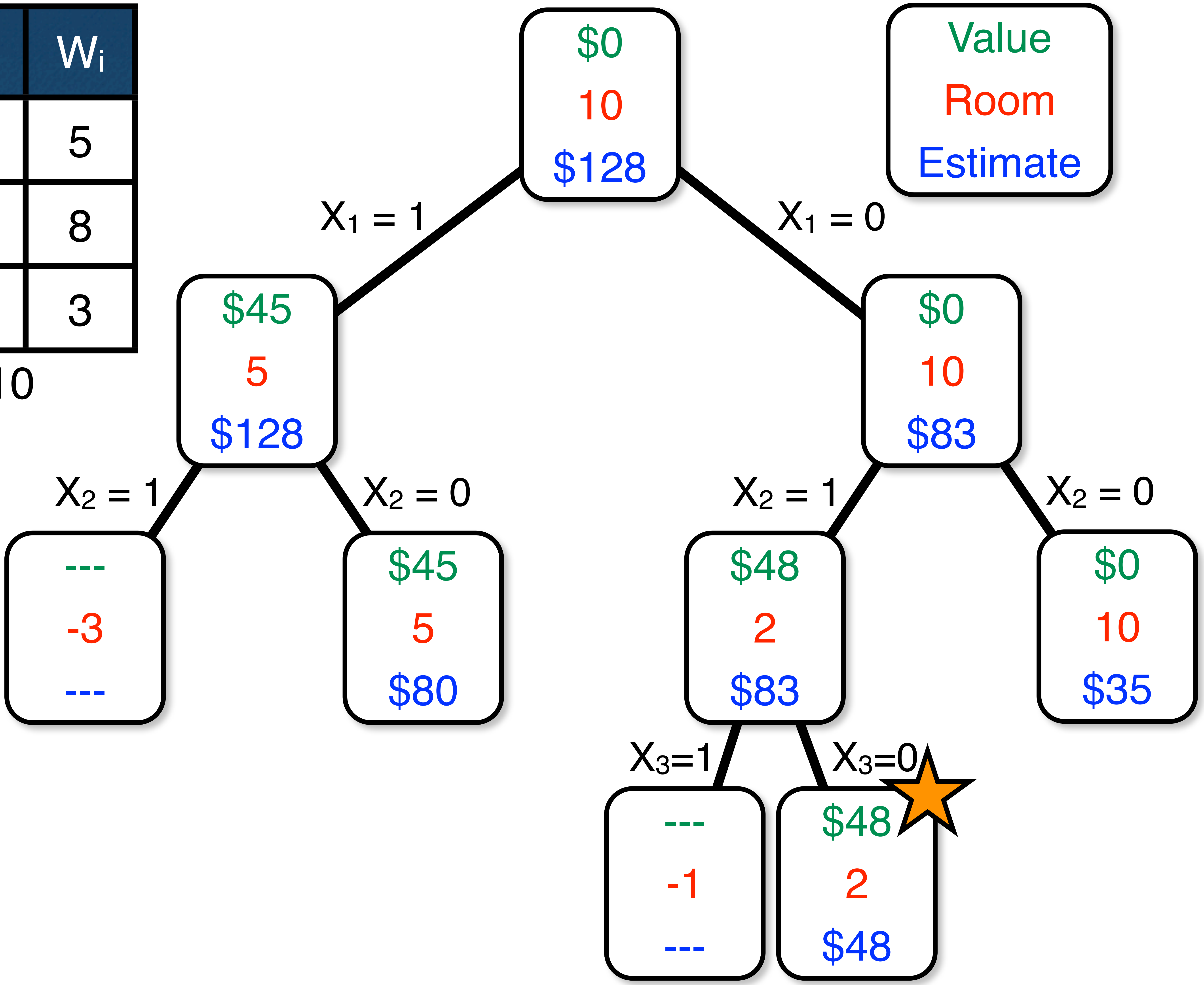
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

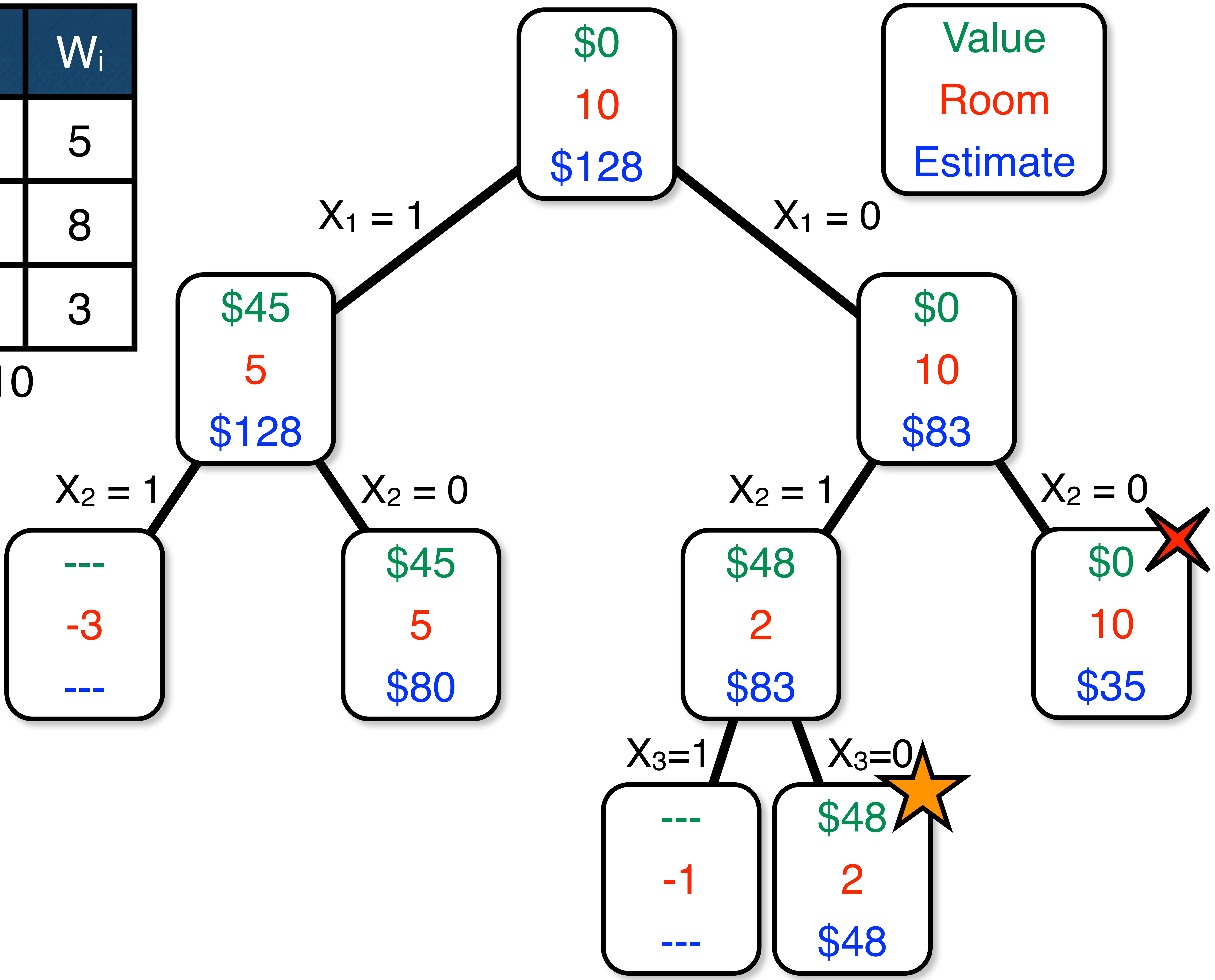
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

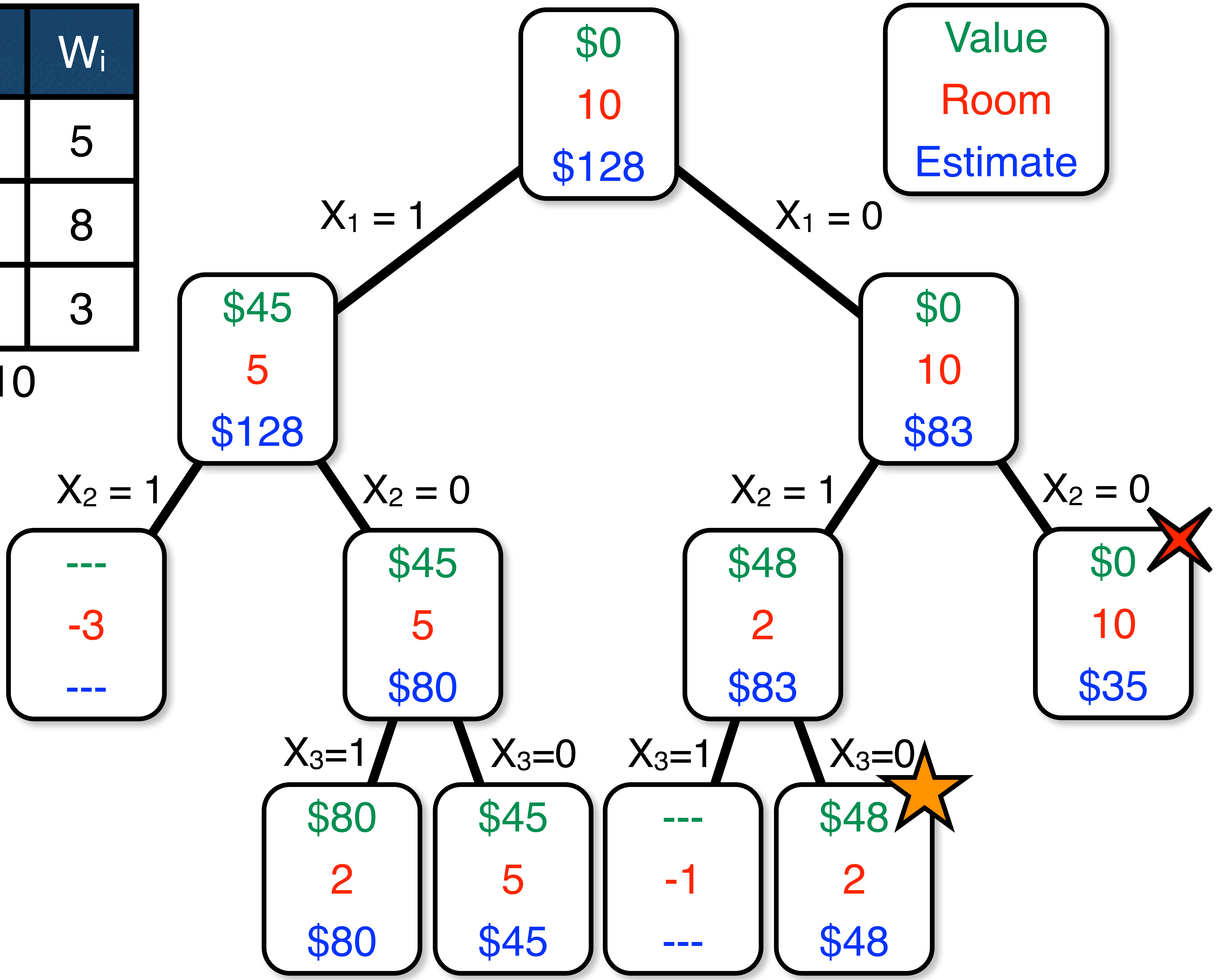
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

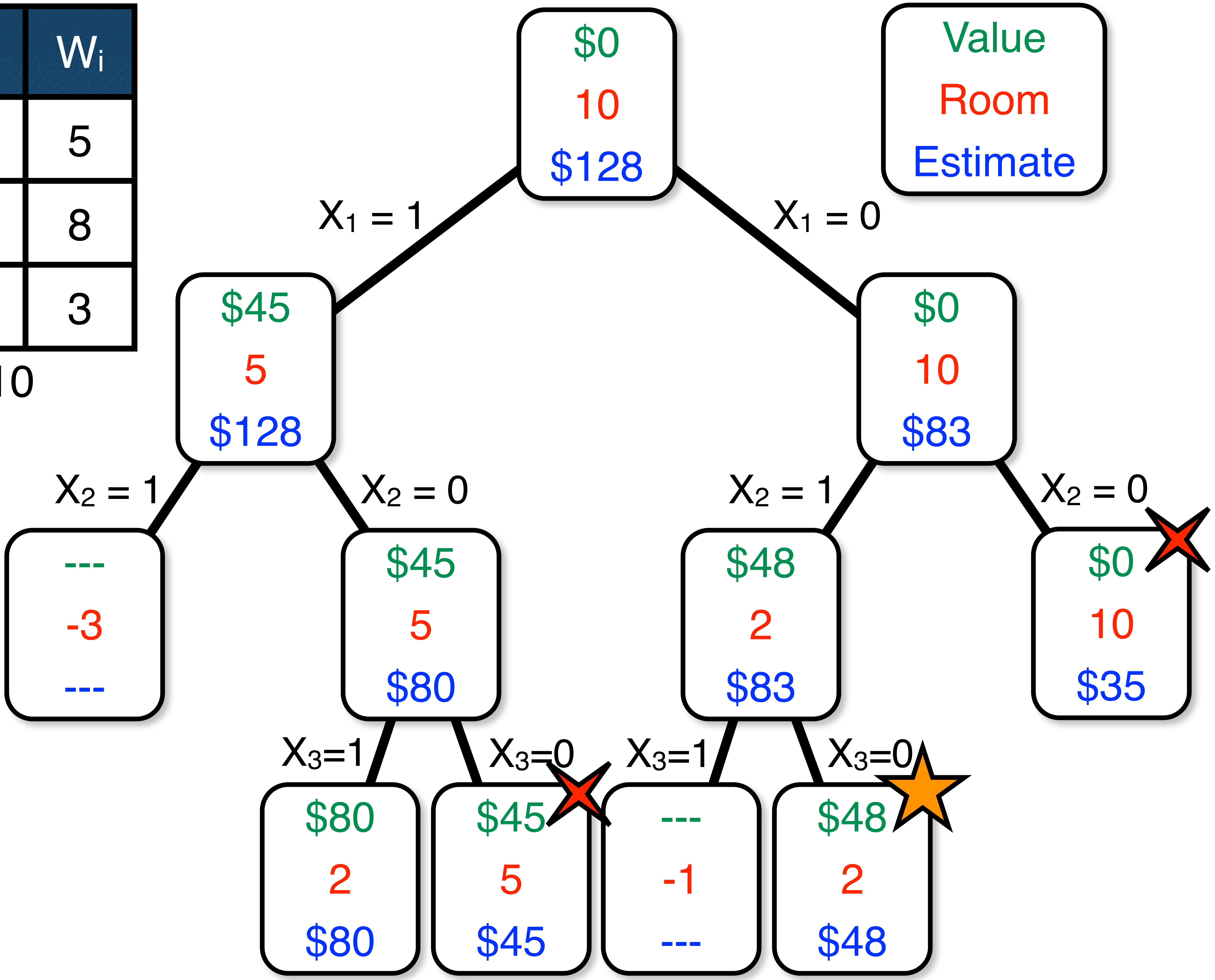
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

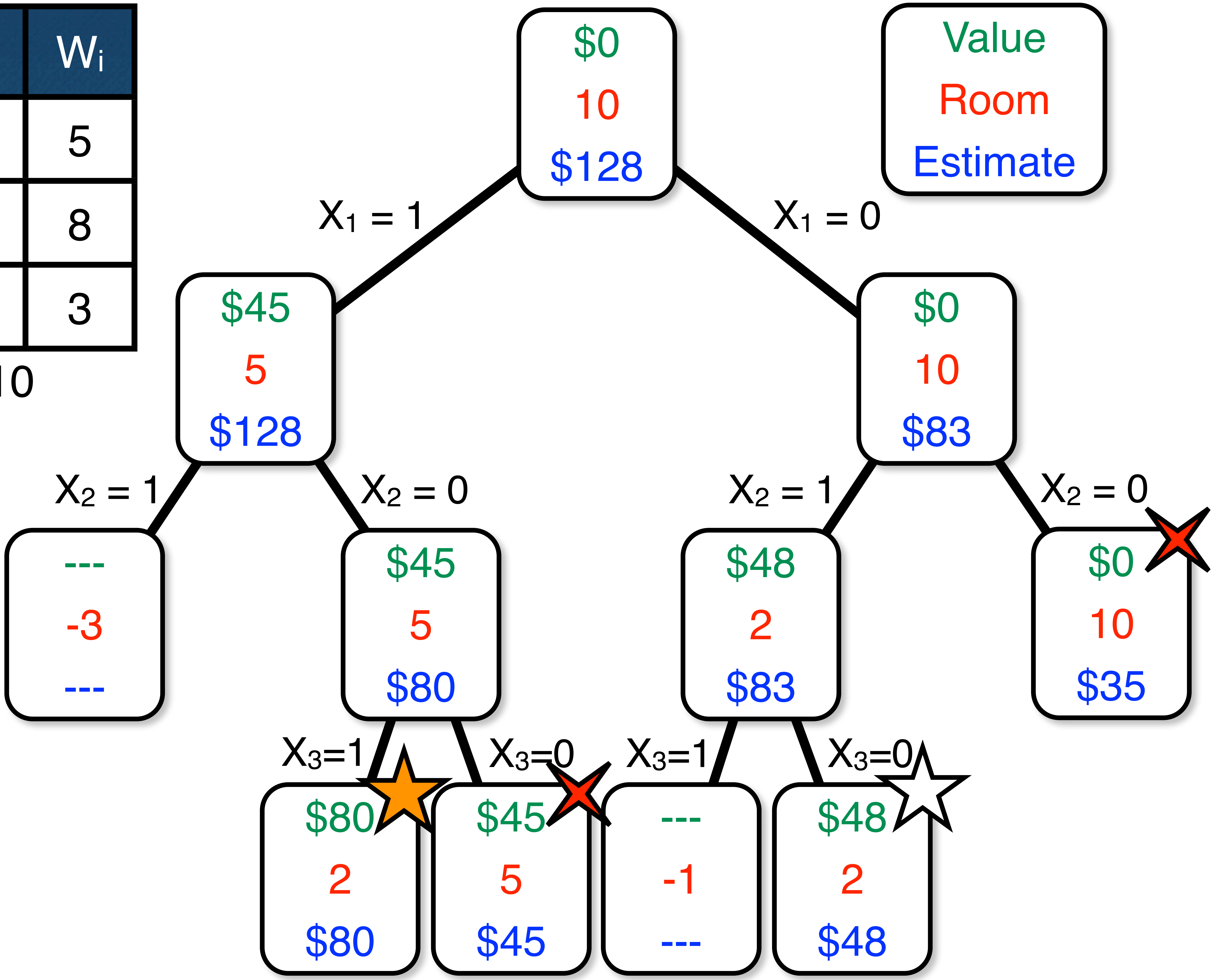
K = 10



Best-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10



Branch and Bound

- Best-First
  - select the node with the best estimation
  - when does it prune?
    - when all the nodes are worse than a found solution
  - is it memory efficient?
    - exaggerate!

Dynamic Programming and Branch and Bound

- Which one is better?

Dynamic Programming and Branch and Bound

- Which one is better?

- Can you combine the two?

Until Next Time