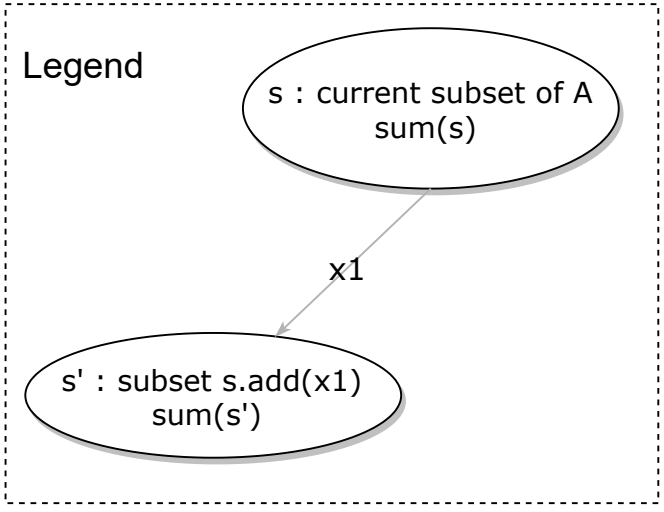
Gift Coupon (Sum of Subset Problem)

State-Space Search Method : Enumerating all subsets with DFS

Problem:
A = [3, 4, 2, 5]
P = 6



State-Space Graph กราฟแสดงการค้นหาคำตอบ

ลำดับของการค้นหา (check ค่า sum)
วิธีการ Depth-First Search
(pre-order processing: node-left-right)

1: [ ] sum = 0
2: [3] sum = 3
3: [3,4] sum = 7
4: [3,4,2] sum = 9
5: [3,4,2,5] sum = 14
6: [3,4, 5] sum = 12
7: [3, 2] sum = 5
8: [3, 2, 5] sum = 10
9: [3, 5] sum = 8
10: [4] sum = 4
11: [4, 2] sum = 6

***** sum = P = 6 *****

Found Solution :

sum([4,2]) = 6

จำนวนโหนดที่ค้น = 11 โหนด

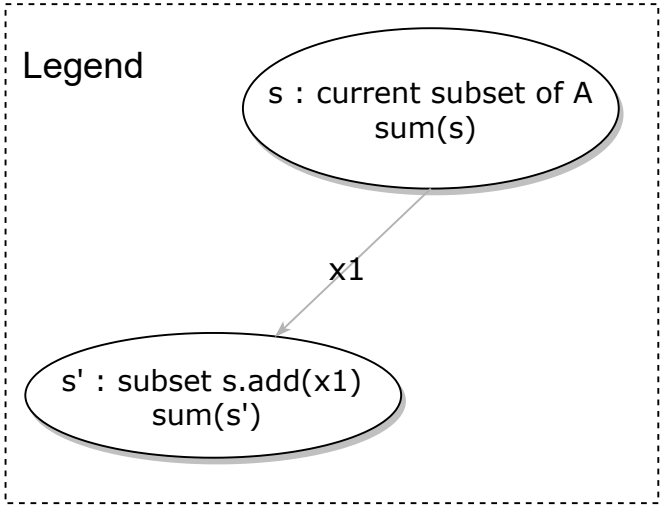
Gift Coupon (Sum of Subset Problem)

State-Space Search Method : Enumerating all subsets with Breadth-First Search (BFS)

Problem:

A = [3, 4, 2, 5]

P = 6



State-Space Graph กราฟแสดงการค้นหาคำตอบ

ลำดับของการค้นหา (check ค่า sum)
วิธีการ Breadth-First Search
(processing order: level by level, left-to-right)

1: [ ] sum = 0
2: [3] sum = 3
3: [4] sum = 4
4: [2] sum = 2
5: [5] sum = 5
6: [3, 4] sum = 7
7: [3, 2] sum = 5
8: [3, 5] sum = 8
9: [4, 2] sum = 6

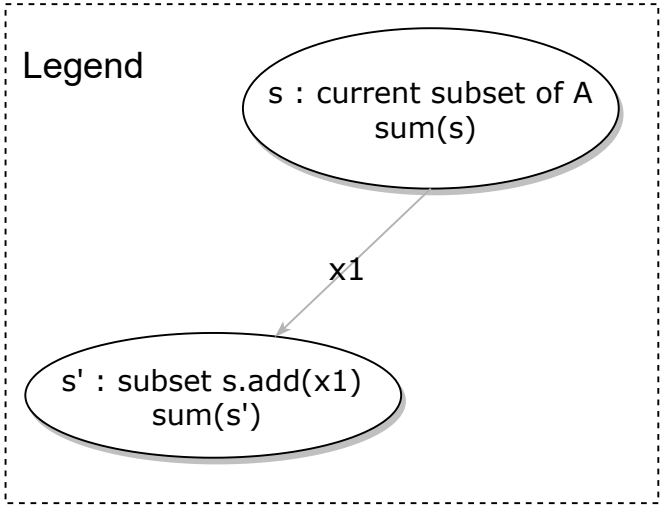
***** sum = P = 6 *****
Found Solution :
sum([4,2]) = 6

จำนวนโหนดที่ค้น = 9 โหนด

Gift Coupon (Sum of Subset Problem)

State-Space Search Method :
Enumerating all subsets with Backtracking

Problem:
A = [3, 4, 2, 5]
P = 6



State-Space Graph กราฟแสดงการค้นหาคำตอบ

ลำดับของการค้นหา (check ค่า sum)
วิธีการ Backtracking
Checking condition :
If sum > P then Backtrack (cutting all the subtrees / pruning)

1: [ ] sum = 0 <P
2: [3] sum = 3 <P
3: [3,4] sum = 7 >P **Backtrack
4: [3, 2] sum = 5 <P
5: **[3, 2, 5] sum = 10 > P (leaf node)**
6: **[3, 5] sum = 8 > P (leaf node)**
7: [4] sum = 4 <P
8: [4, 2] sum = 6 = P

***** sum = P = 6 *****
Found Solution :
sum([4,2]) = 6

จำนวนโหนดที่ค้น = 8 โหนด

โหนดที่ถูกตัดไป ไม่ต้องถูกค้น 3 โหนด
[3,4,2] [3,4,2,5] [3,4,5]

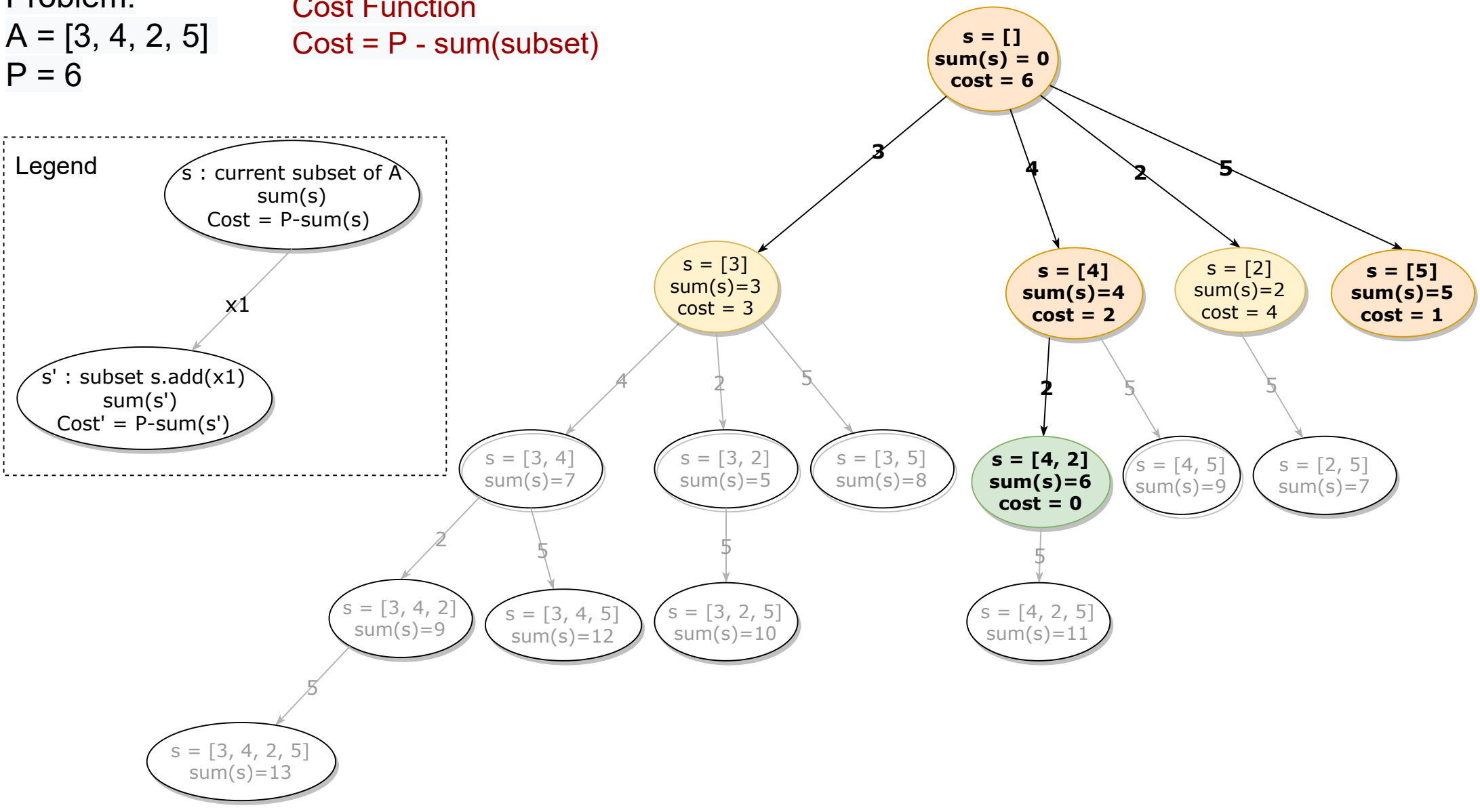
Gift Coupon (Sum of Subset Problem)

State-Space Search Method : Least-Cost Search (LCS)

Problem:
A = [3, 4, 2, 5]
P = 6

Cost Function
Cost = P - sum(subset)

State-Space Graph กราฟแสดงการค้นหาคำตอบ



ลำดับของการค้นหา (check ค่า sum)
วิธีการ Least-Cost Search
(processing order by least cost)

1: [ ] sum = 0 , cost = 6 , sum != P
===> generate children nodes [3], [4], [2], [5]
2: [3] sum = 3, cost = 3
3: [4] sum = 4, cost = 2
4: [2] sum = 2, cost = 4
5: [5] sum = 5, cost = 1 (least cost)
[5] ==> sum != P, No child node
[4] cost = 2 (least cost) sum !=P
===> generate children nodes [4,2], [4,5]
6: [4, 2] sum = 6, cost = 0 ** sum = P
***** **sum = P = 6** *******
Found Solution :
sum([4,2]) = 6

จำนวนโหนดที่ค้น = 6 โหนด