

FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

Projeto e Análise de Algoritmos

Força Bruta – Branch-and-Bound

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

Branch-and-Bound

- Possui todos os elementos do *backtracking*, mas ele não termina simplesmente ao achar a primeira solução
 - Ele continua até a melhor solução ser encontrada
- Ele armazena a pontuação da solução (estado) mais promissora

Branch-and-Bound

- Inicia novas buscas pela ramificação mais promissora
 - Aquela que maximiza ou minimiza a função objetivo
- É capaz de podar ramificações
 - Por exemplo, se procurar um caminho mínimo, é possível descartar uma ramificação que excederá um caminho mínimo já conhecido

Branch-and-Bound

- Exemplo
 - Problema da Associação de Tarefas (Assignment Problem)

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

□

Custo: 0

Solução: □

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 5

[1,0,0,0]

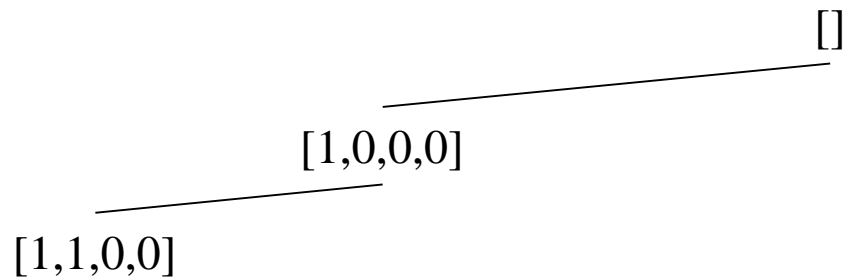
[]

Solução: []

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 10

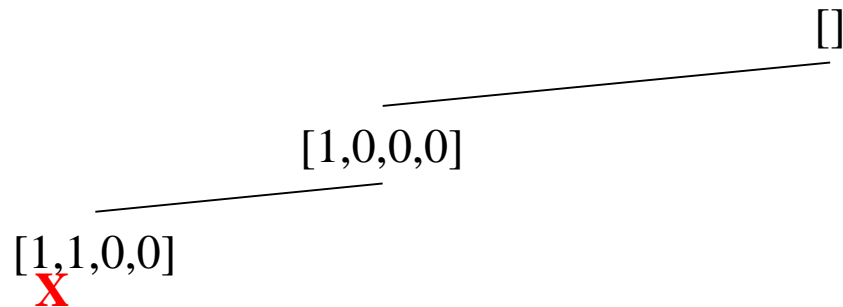


Solução: []

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 10

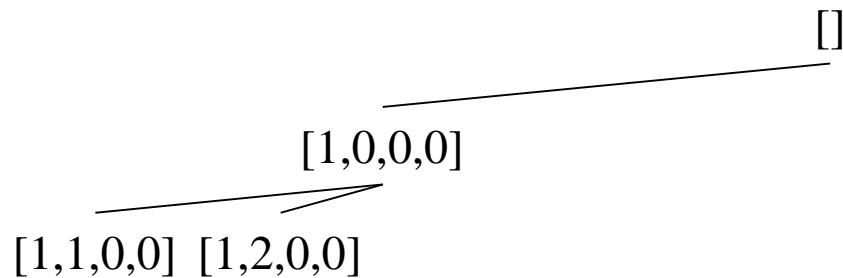


Solução: $[\]$

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 6

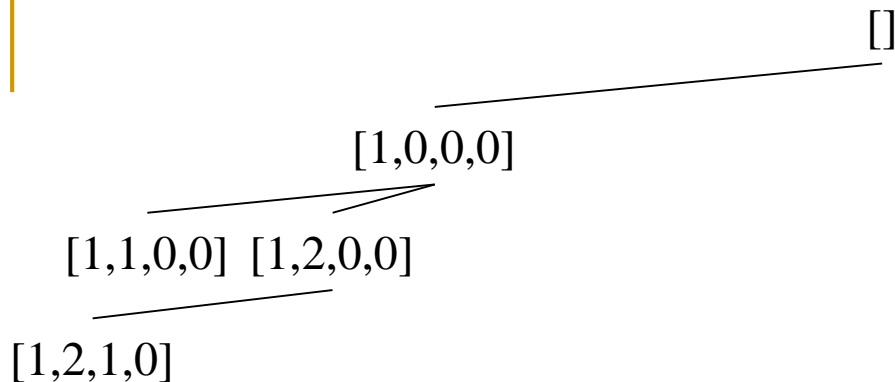


Solução: $[\]$

Custo: $+\text{inf}$

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

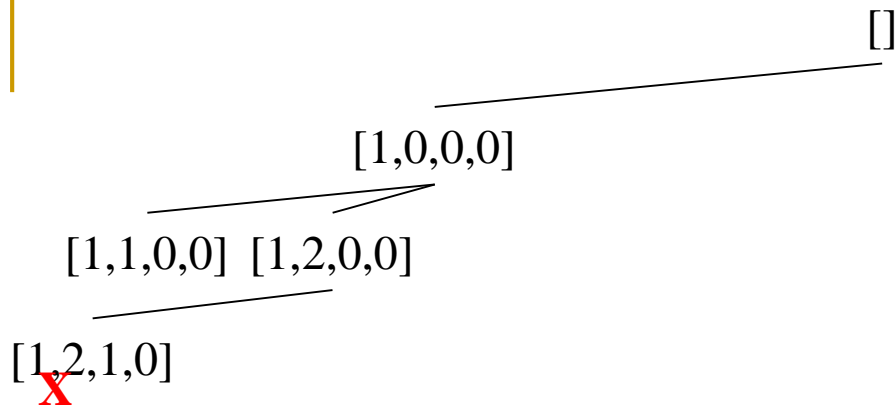
Custo: 12



Solução: []

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

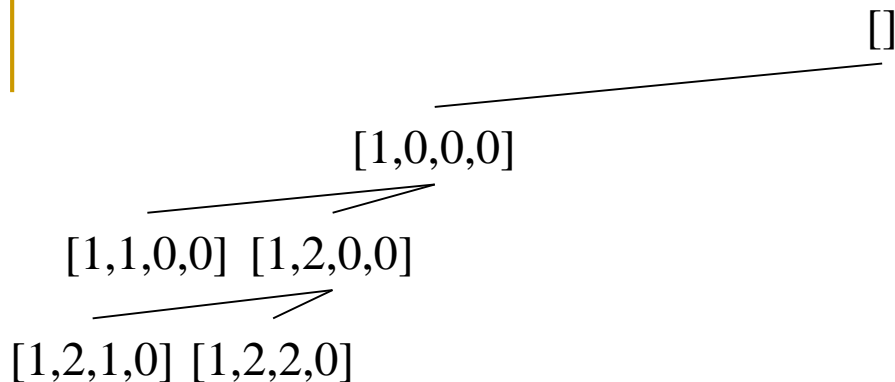


Solução: $[]$

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 10

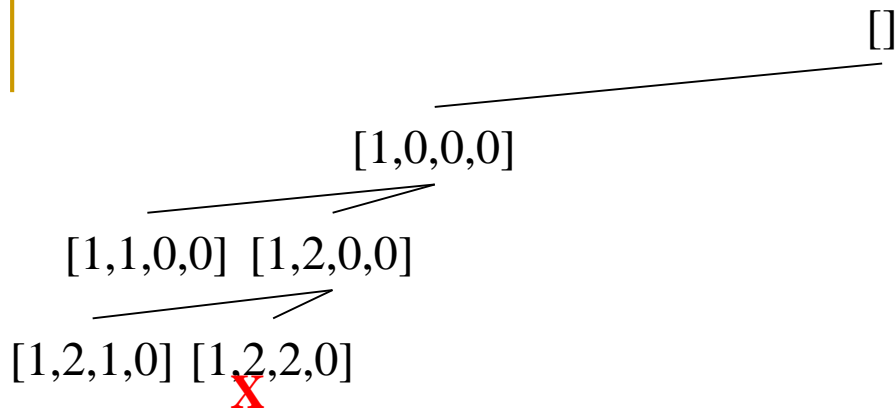


Solução: []

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

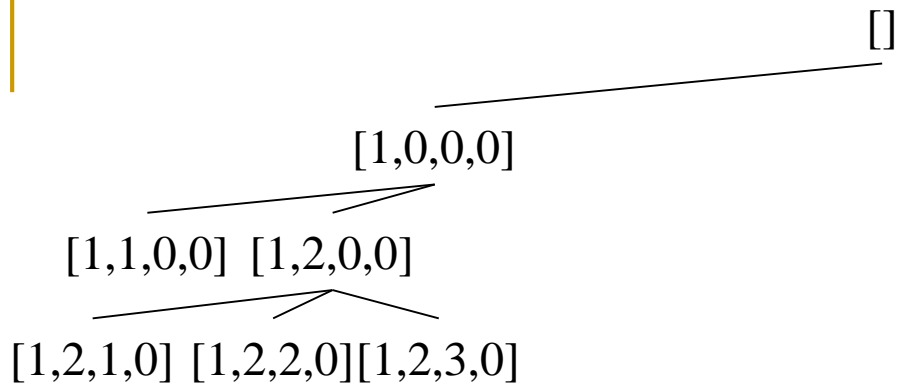
Custo: 10



Solução: []

Custo: +inf

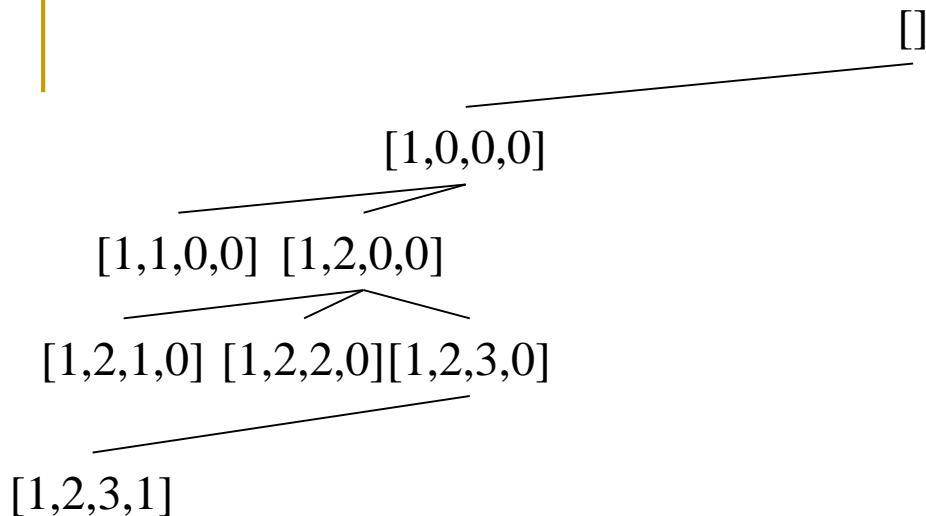
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: $[\]$

Custo: $+\text{inf}$

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

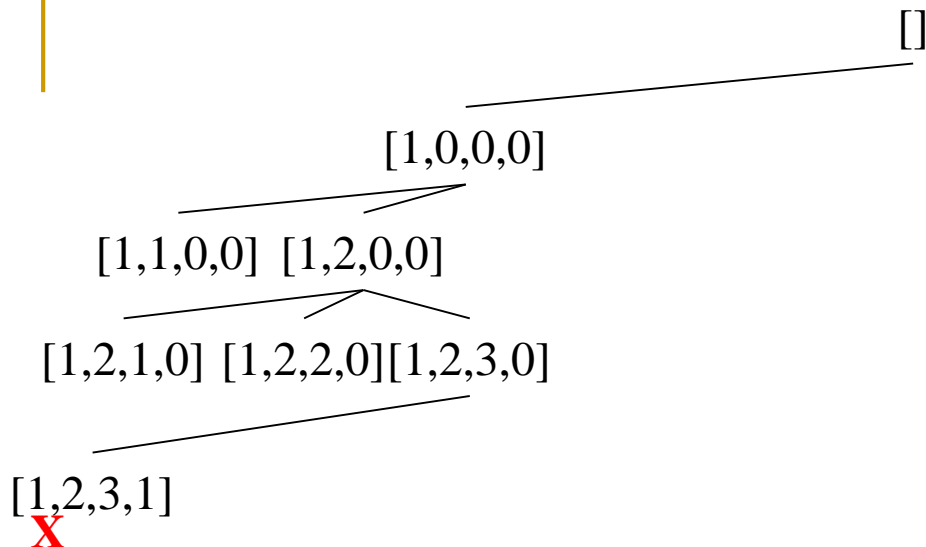


Solução: []

Custo: +inf

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

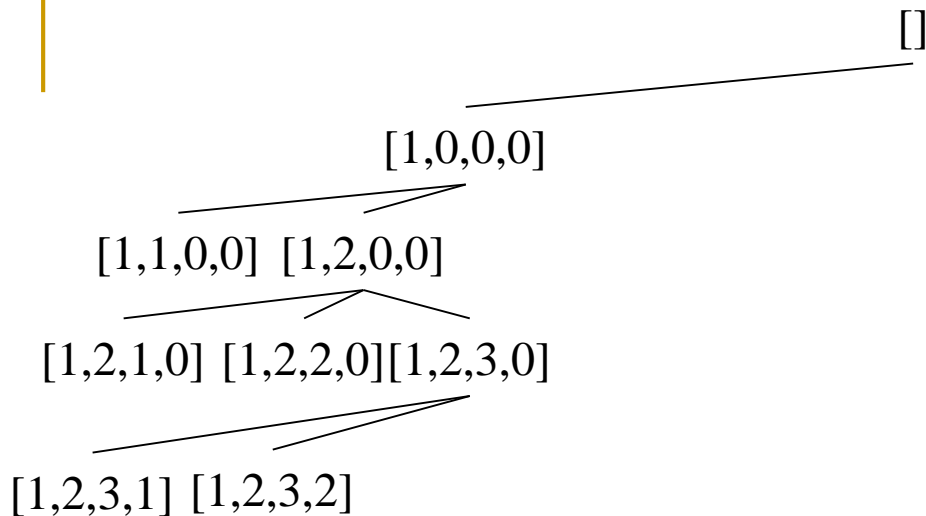
Custo: 28



Solução: $[\]$

Custo: $+\text{inf}$

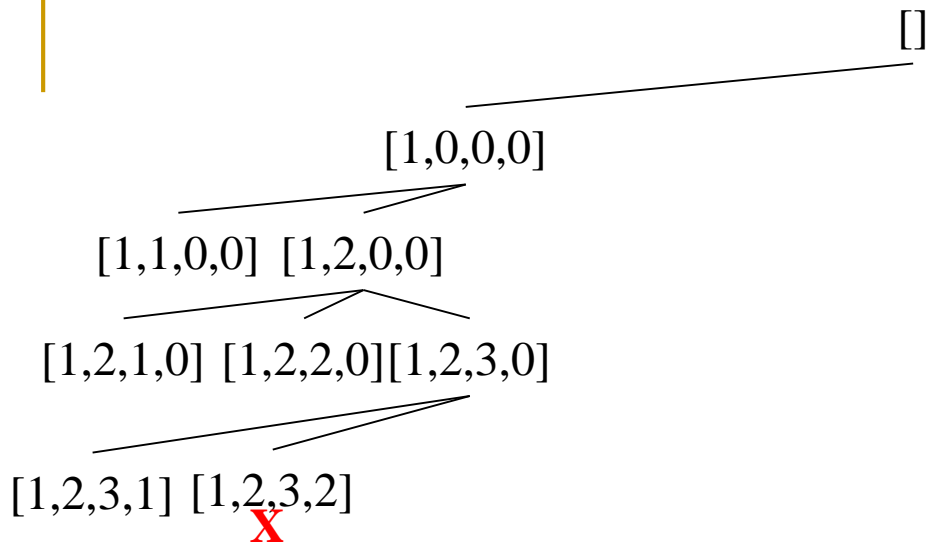
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: []

Custo: +inf

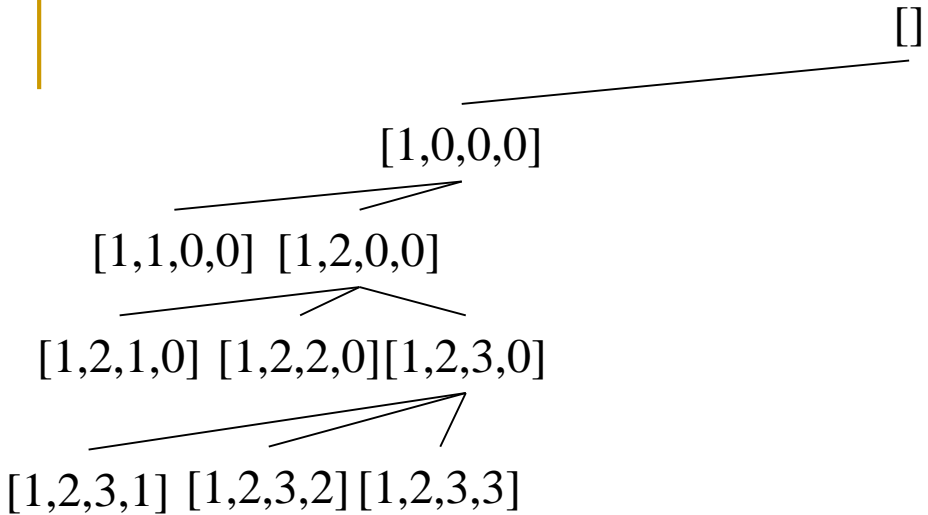
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: []

Custo: +inf

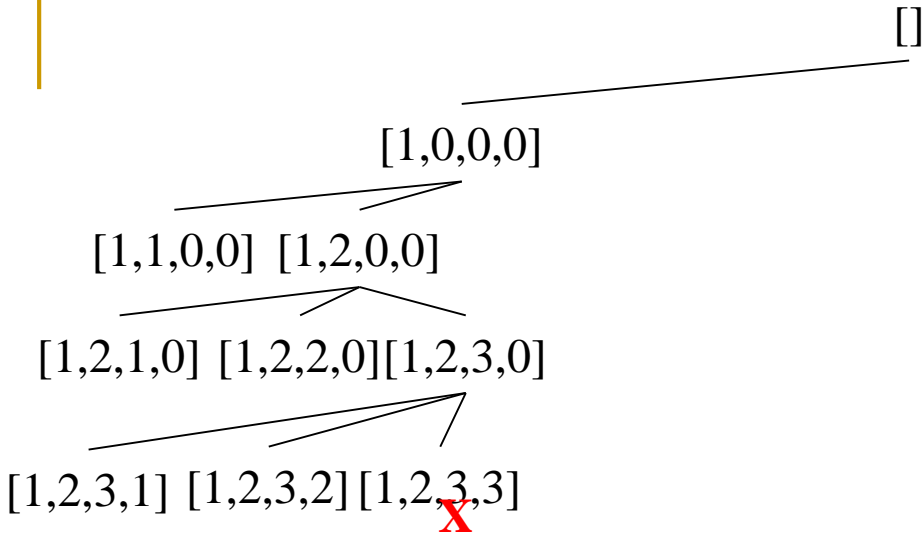
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: []

Custo: +inf

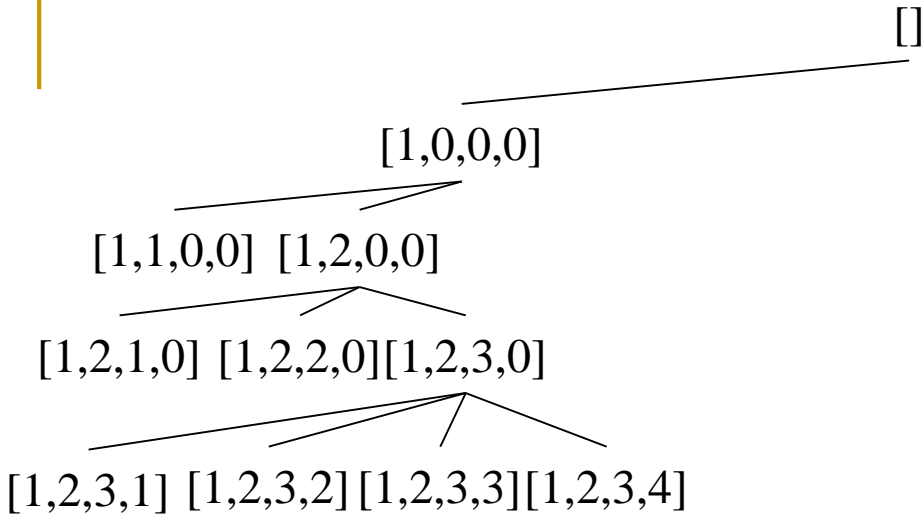
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: []

Custo: +inf

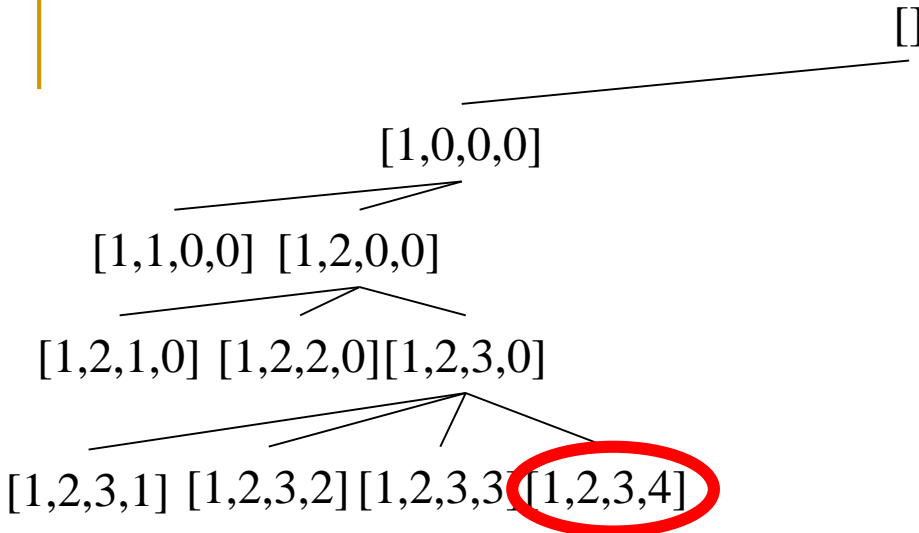
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: []

Custo: +inf

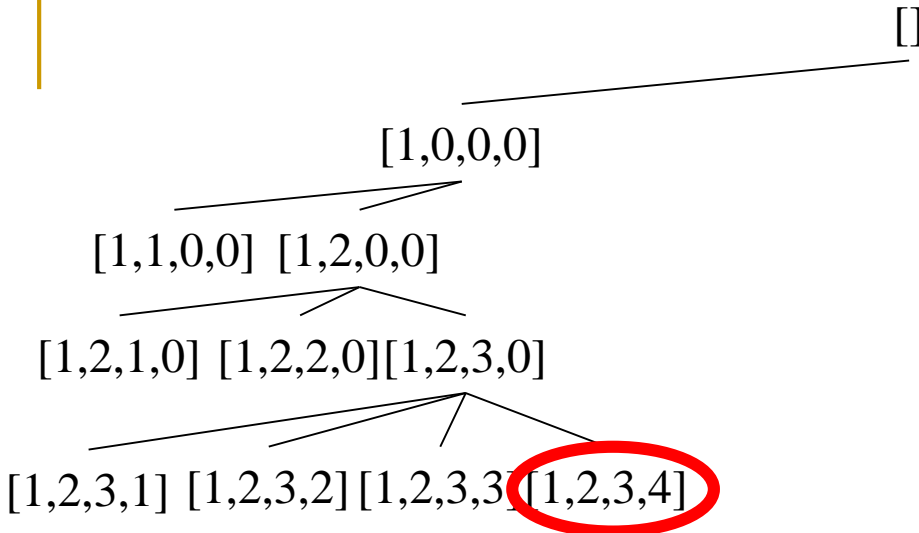
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Custo da solução atual: +inf
Custo da nova solução: 29

Solução: []
Custo: +inf

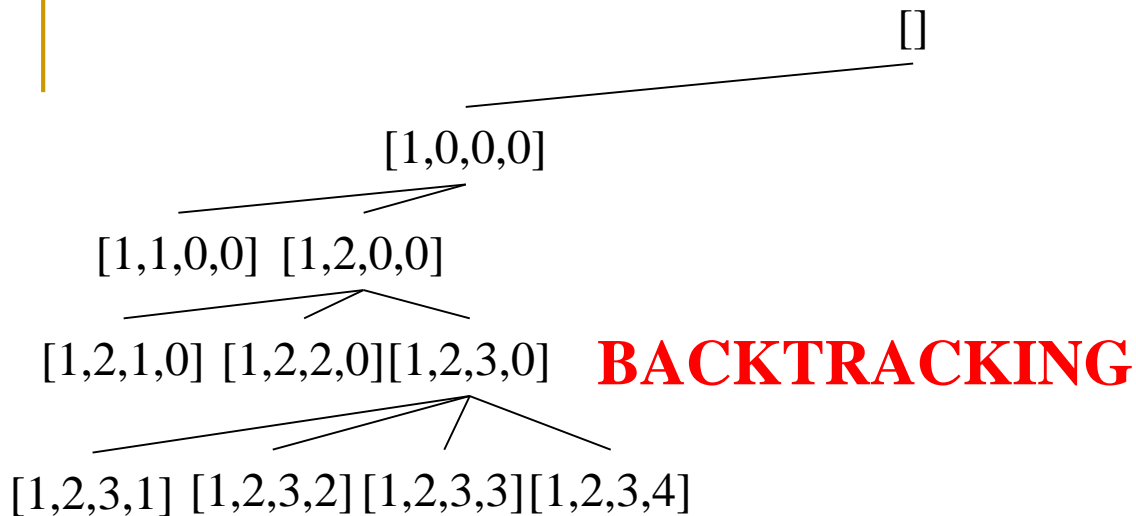
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 3, 4]

Custo: 29

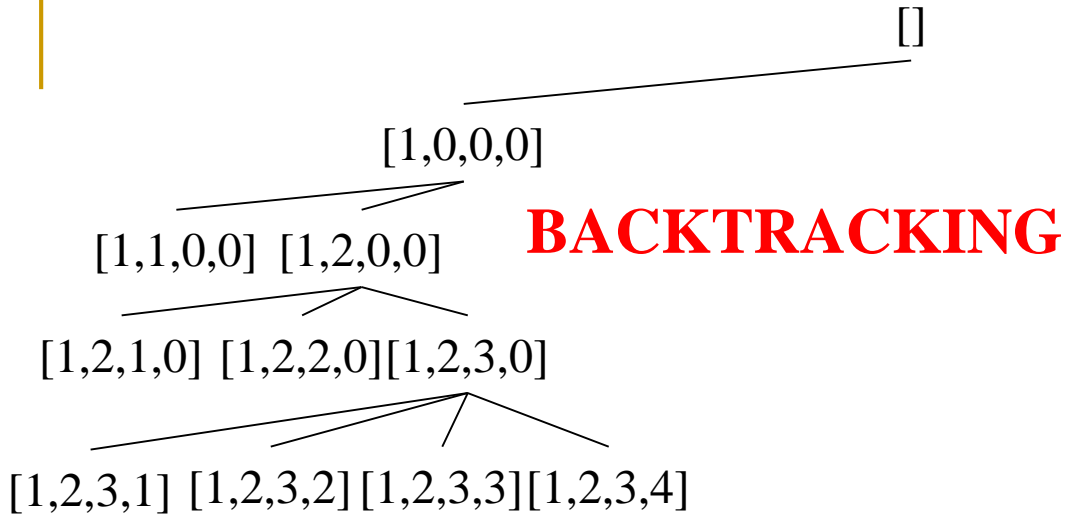
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 3, 4]

Custo: 29

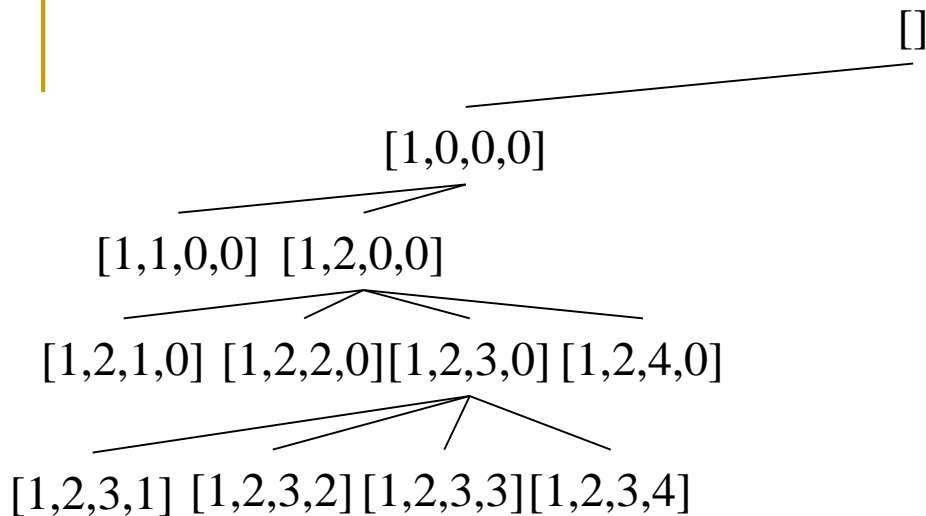
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 3, 4]

Custo: 29

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 3, 4]

Custo: 29

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

[]

[1,0,0,0]

[1,1,0,0] [1,2,0,0]

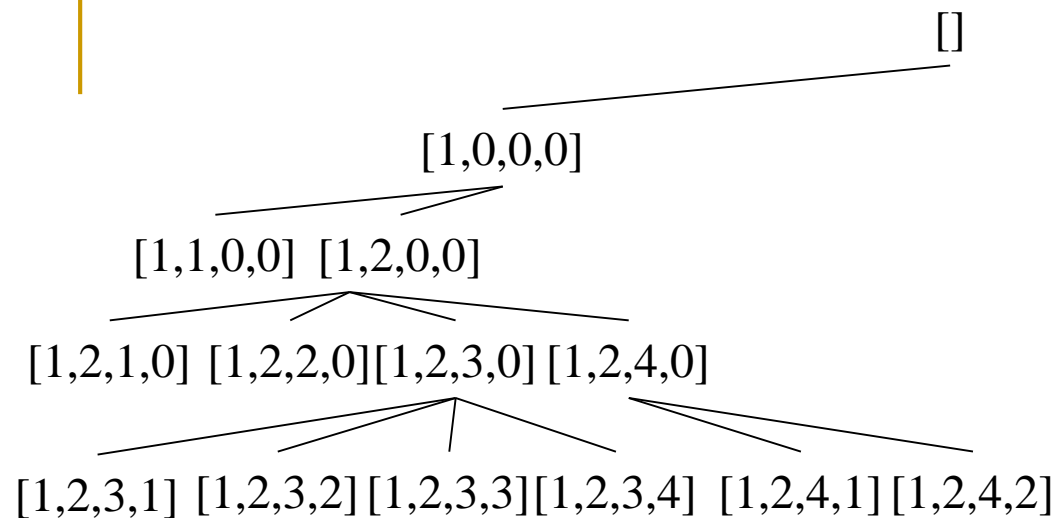
[1,2,1,0] [1,2,2,0][1,2,3,0] [1,2,4,0]

[1,2,3,1] [1,2,3,2][1,2,3,3][1,2,3,4] [1,2,4,1]

Solução: [1, 2, 3, 4]

Custo: 29

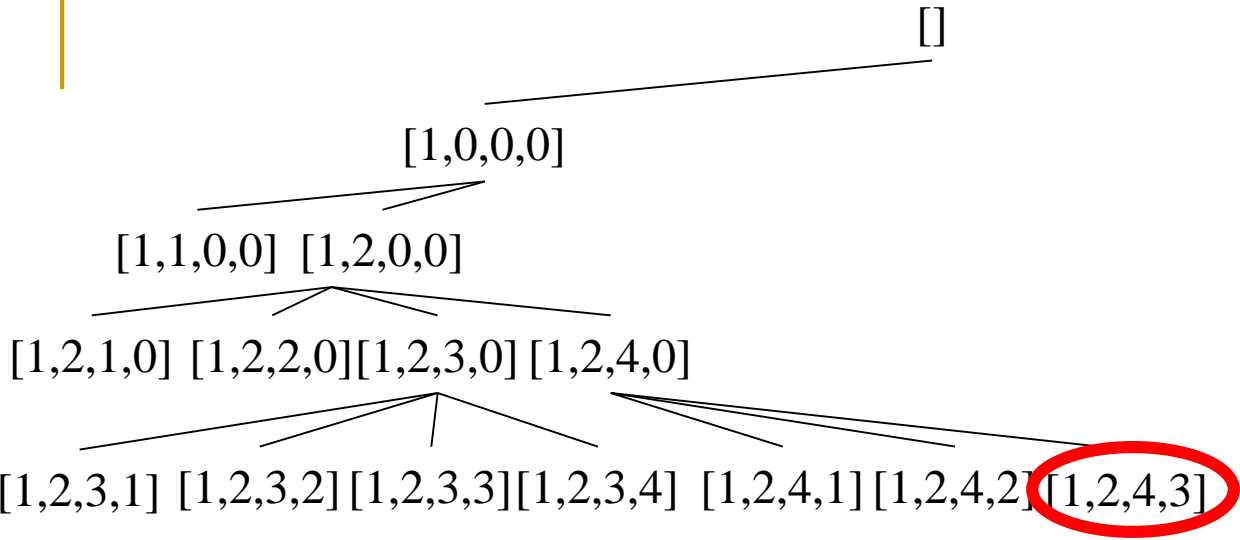
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 3, 4]

Custo: 29

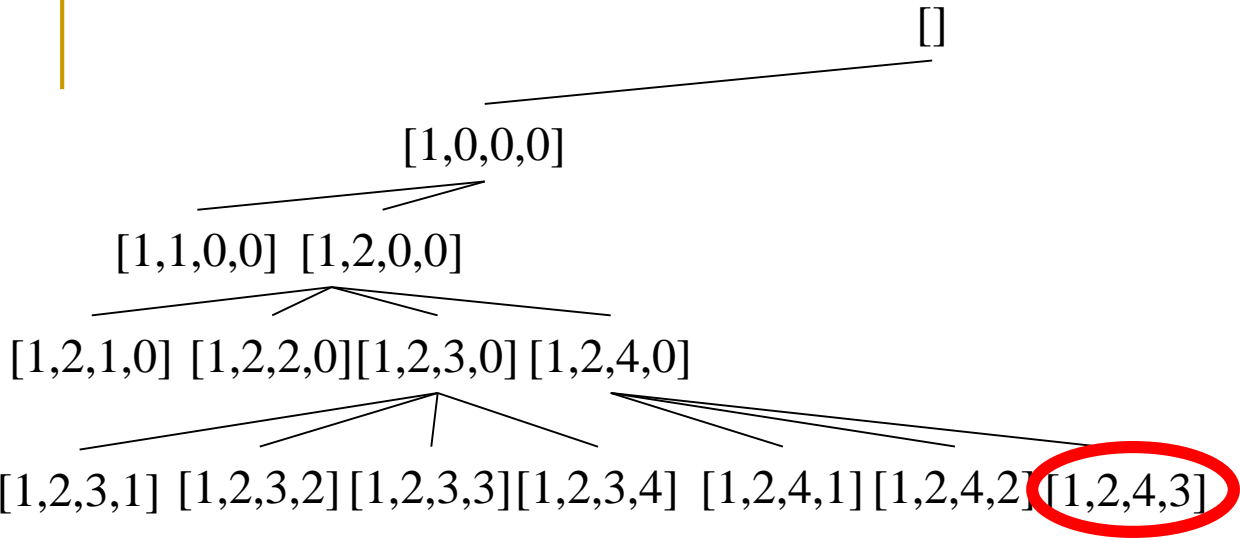
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Custo da solução atual: 29
Custo da nova solução: 23

Solução: [1, 2, 3, 4]
Custo: 29

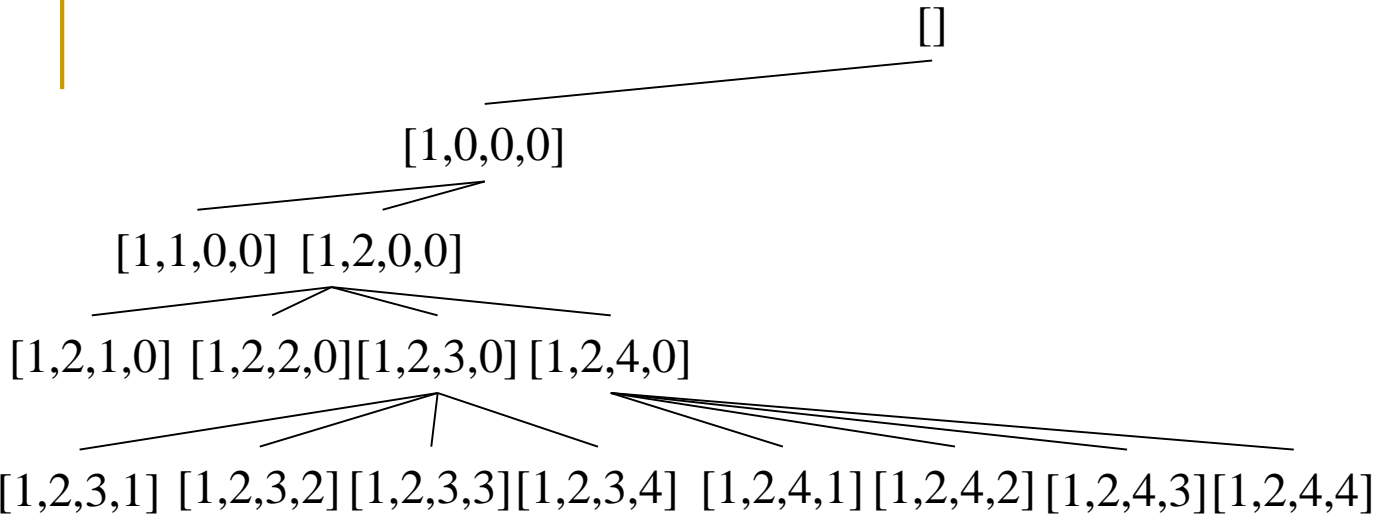
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

[]

[1,0,0,0]

[1,1,0,0] [1,2,0,0]

[1,2,1,0] [1,2,2,0][1,2,3,0] [1,2,4,0]

BACKTRACKING

[1,2,3,1] [1,2,3,2][1,2,3,3][1,2,3,4] [1,2,4,1][1,2,4,2][1,2,4,3][1,2,4,4]

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

[]

[1,0,0,0]

[1,1,0,0] [1,2,0,0]

BACKTRACKING

[1,2,1,0] [1,2,2,0][1,2,3,0] [1,2,4,0]

[1,2,3,1] [1,2,3,2][1,2,3,3][1,2,3,4] [1,2,4,1][1,2,4,2][1,2,4,3][1,2,4,4]

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

[]

[1,0,0,0]

[1,1,0,0] [1,2,0,0]

BACKTRACKING

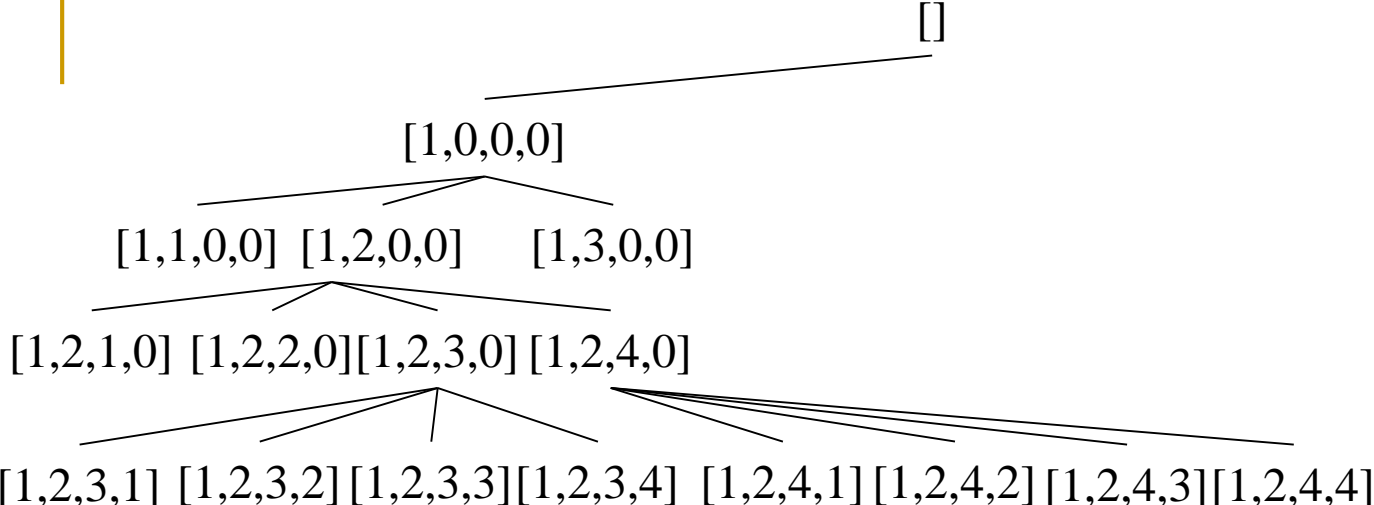
[1,2,1,0] [1,2,2,0][1,2,3,0] [1,2,4,0]

[1,2,3,1] [1,2,3,2][1,2,3,3][1,2,3,4] [1,2,4,1][1,2,4,2][1,2,4,3][1,2,4,4]

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

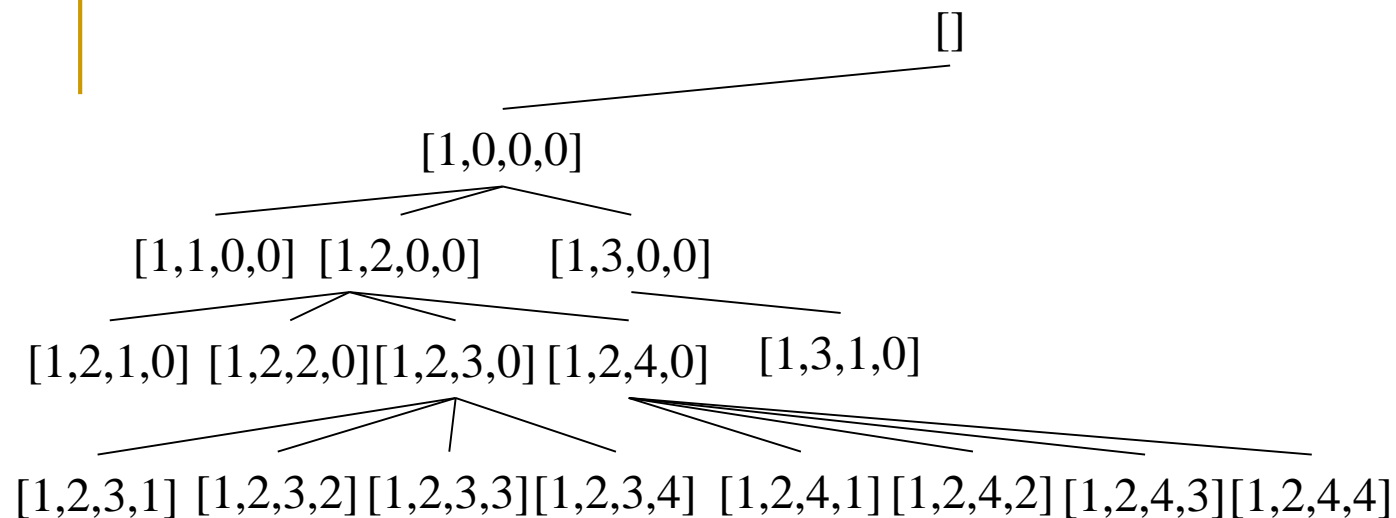


Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

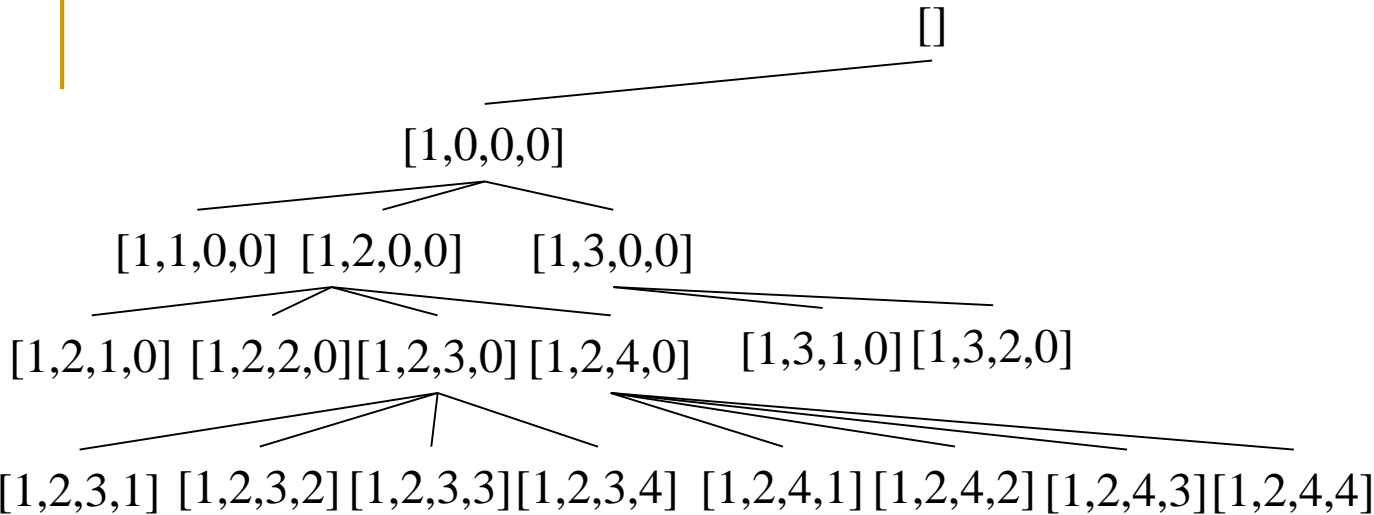
Custo: 16



Solução: [1, 2, 4, 3]

Custo: 23

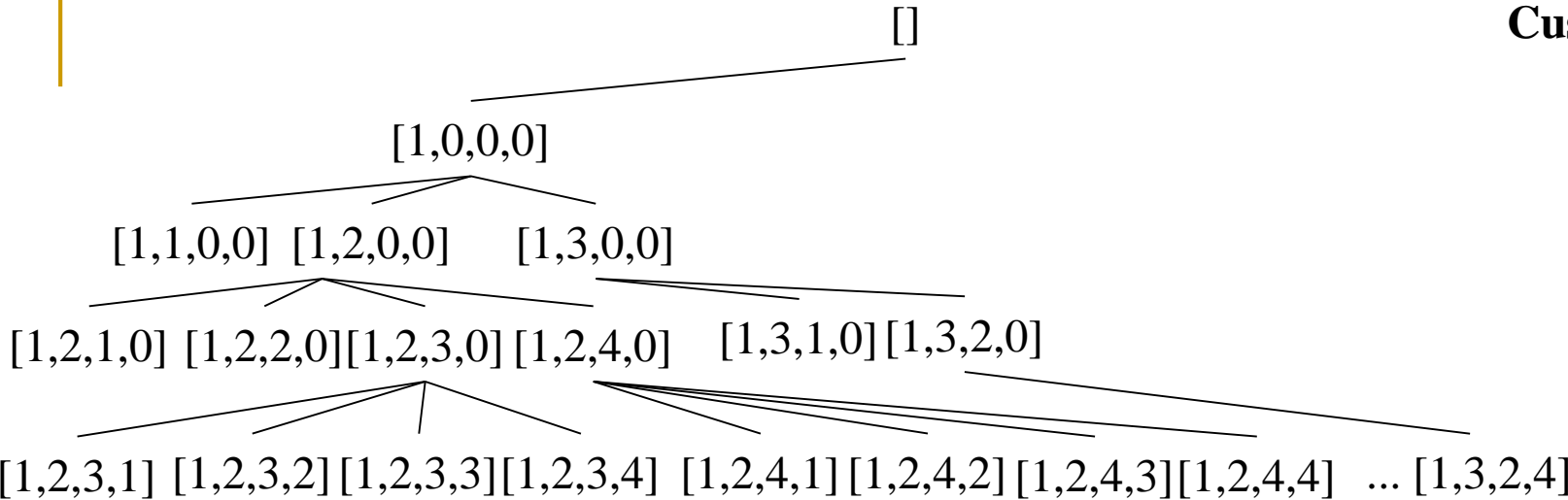
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 4, 3]

Custo: 23

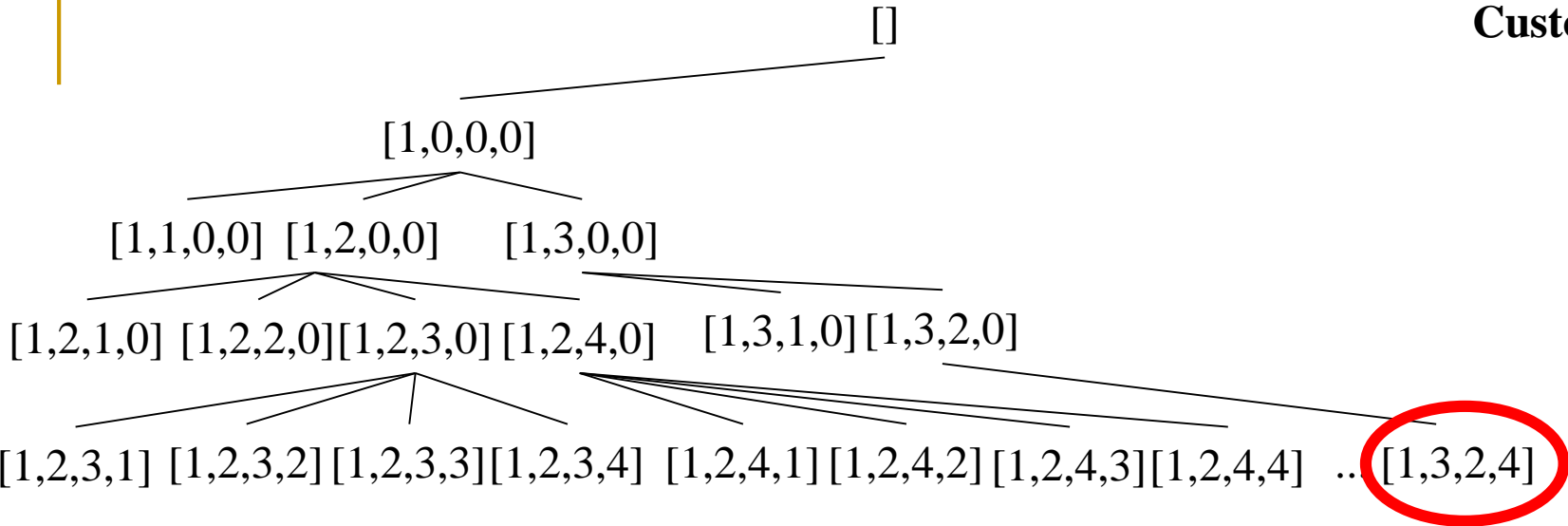
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 4, 3]

Custo: 23

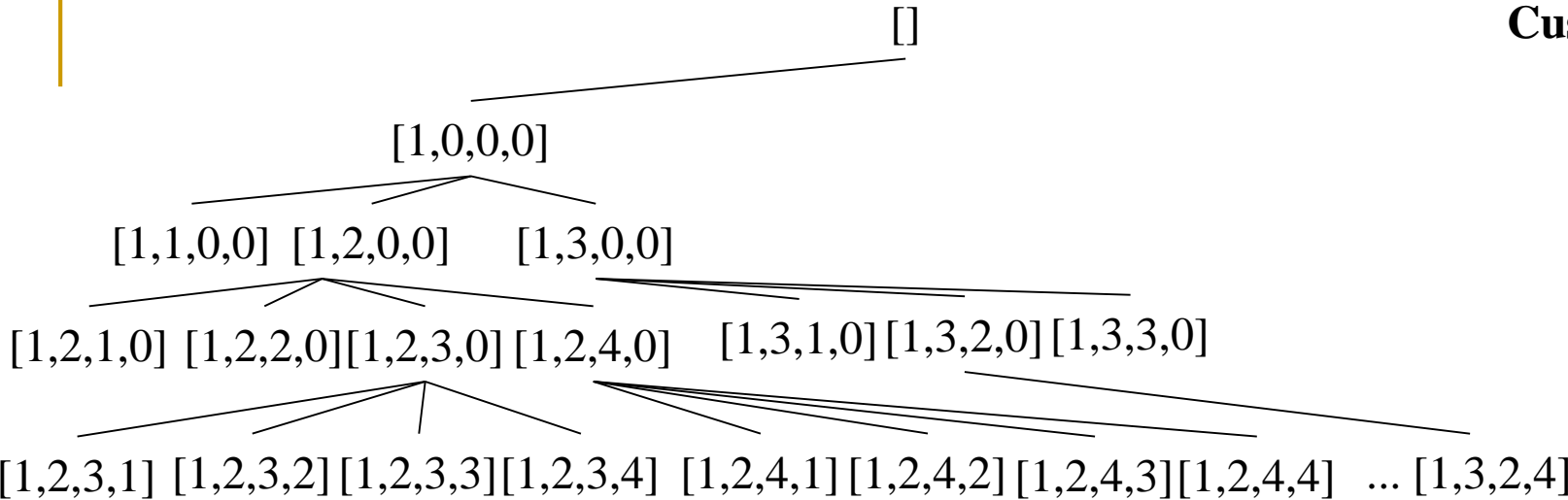
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Custo da solução atual: 23
Custo da nova solução: 24

Solução: [1, 2, 4, 3]
Custo: 23

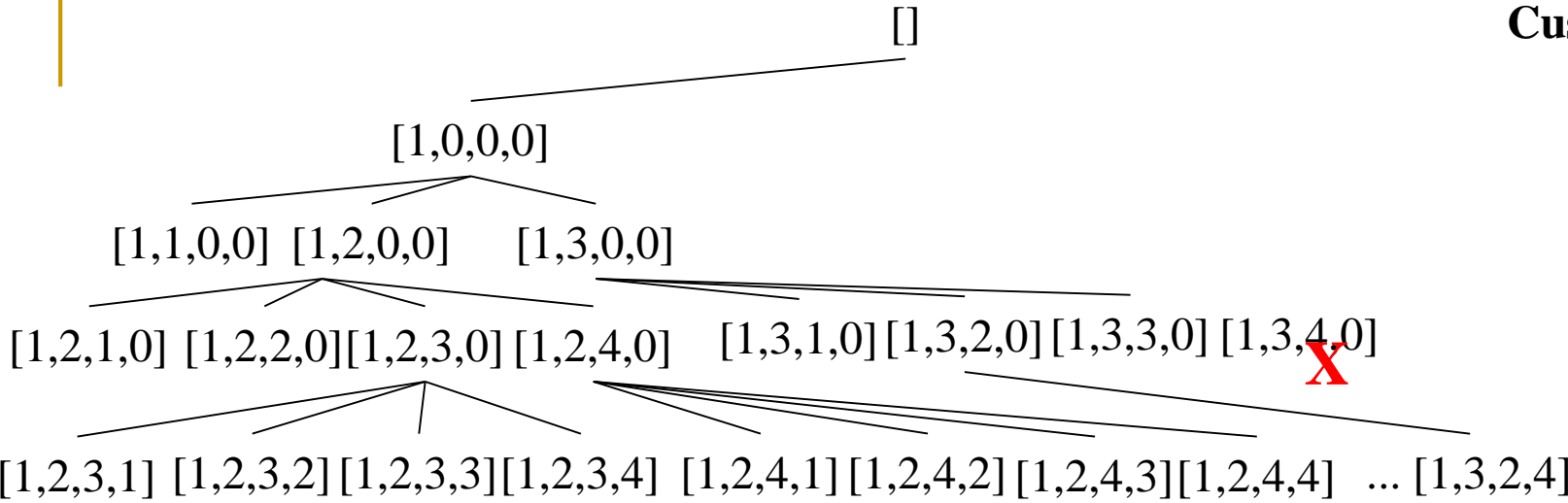
	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

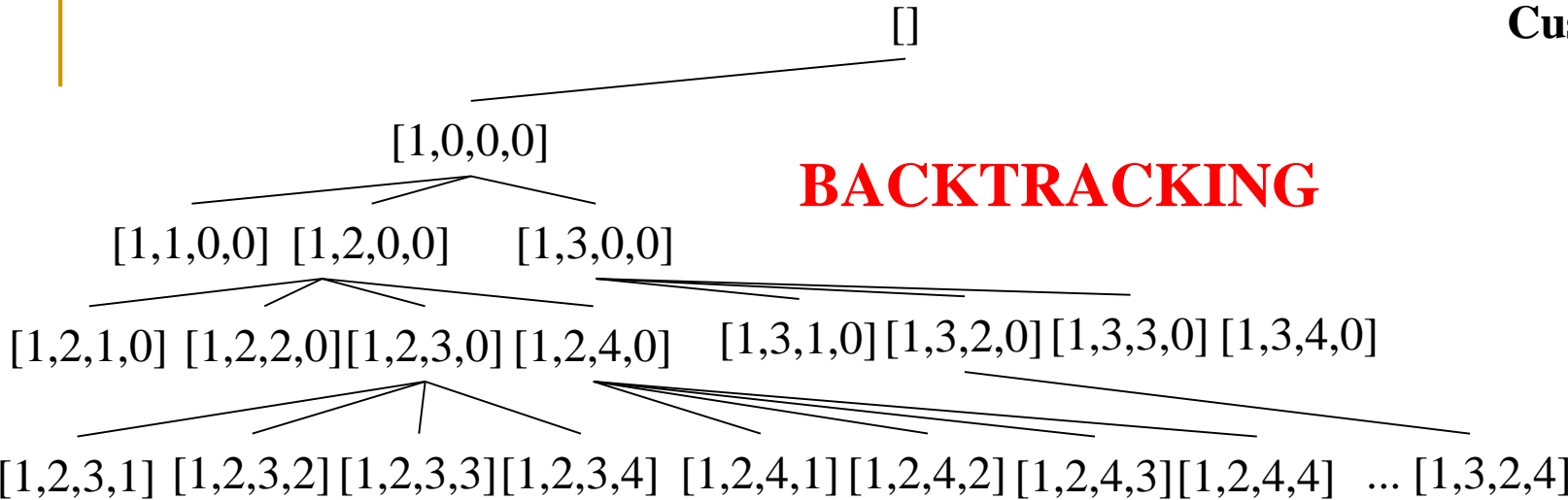


Apesar de ser um estado consistente, ocorre a poda, pois o custo do estado atual é maior do que o custo da solução atual

Solução: [1, 2, 4, 3]
Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

BACKTRACKING



Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

[]

BACKTRACKING

[1,0,0,0]

[1,1,0,0] [1,2,0,0] [1,3,0,0]

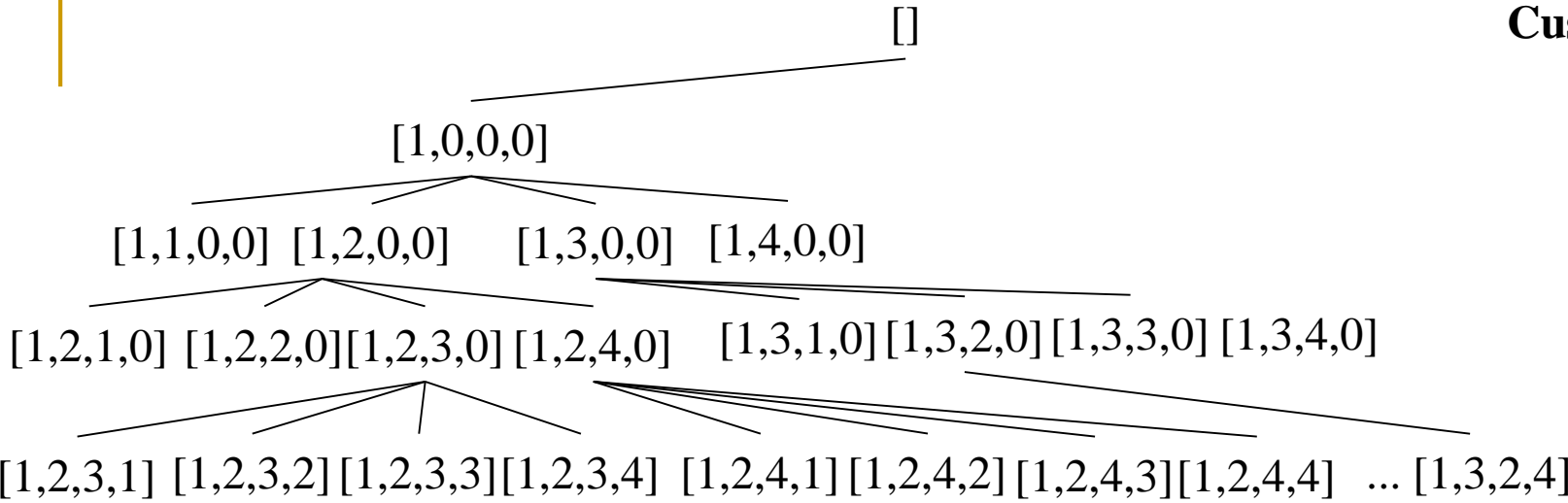
[1,2,1,0] [1,2,2,0] [1,2,3,0] [1,2,4,0] [1,3,1,0] [1,3,2,0] [1,3,3,0] [1,3,4,0]

[1,2,3,1] [1,2,3,2] [1,2,3,3] [1,2,3,4] [1,2,4,1] [1,2,4,2] [1,2,4,3] [1,2,4,4] ... [1,3,2,4]

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

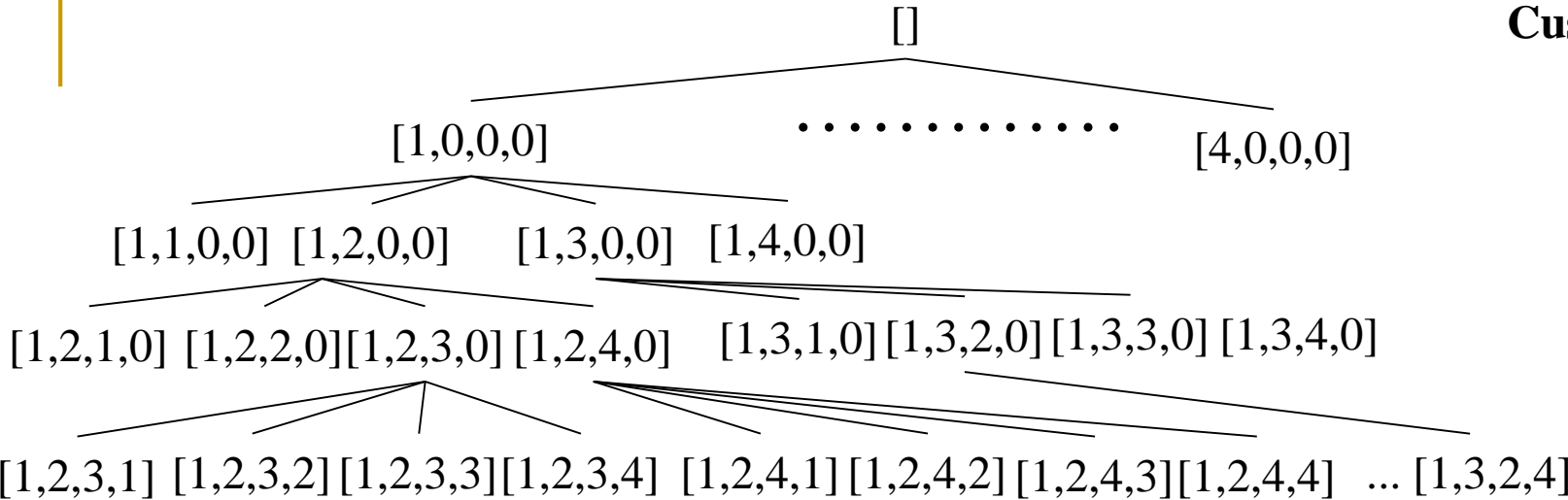


O algoritmo continua até encontrar a melhor solução

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



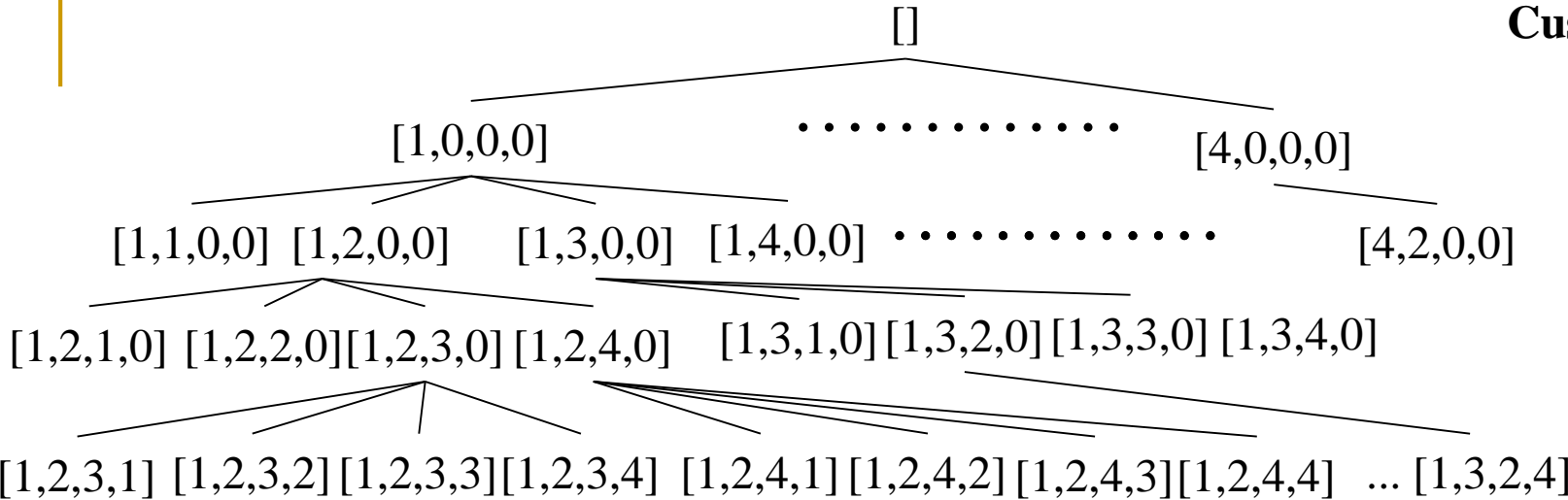
O algoritmo continua até encontrar a melhor solução

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 4

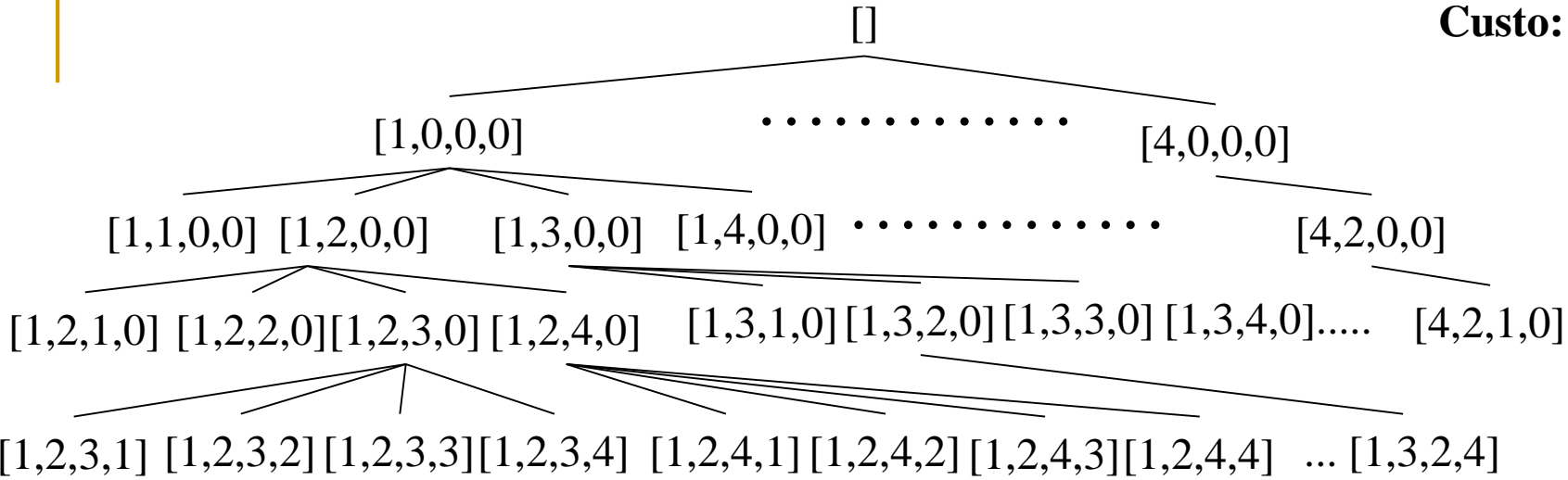


O algoritmo continua até encontrar a melhor solução

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10



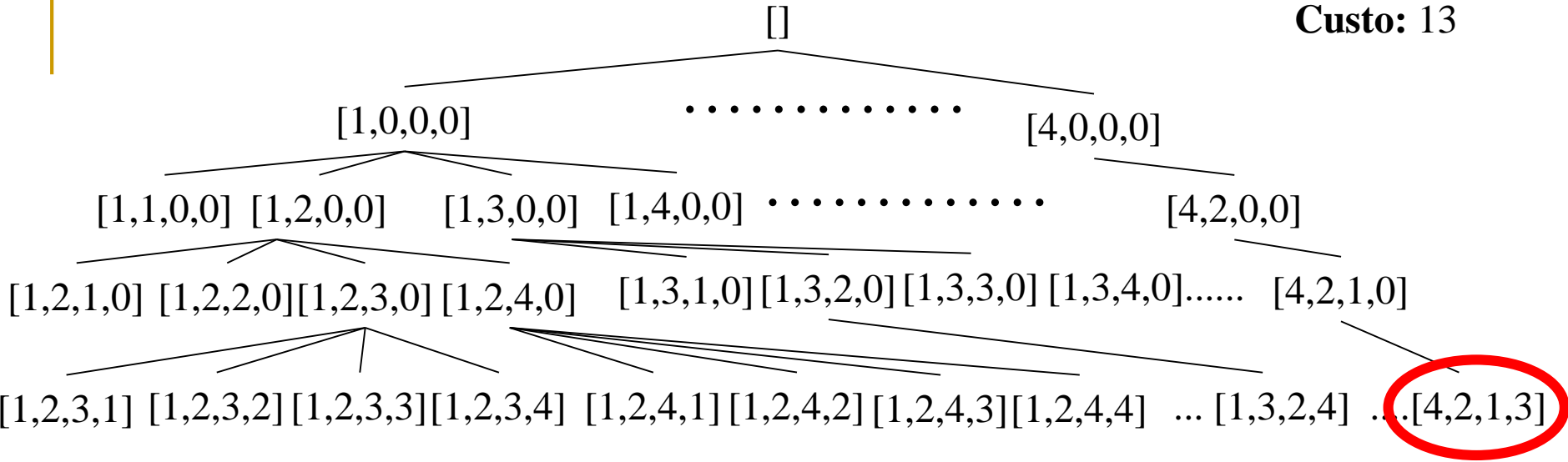
O algoritmo continua até encontrar a melhor solução

Solução: [1, 2, 4, 3]

Custo: 23

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Custo: 13



O algoritmo continua até encontrar a melhor solução

Solução: [4, 2, 1, 3]

Custo: 13

	Tarefa1	Tarefa2	Tarefa3	Tarefa4
Pessoa1	5	5	6	9
Pessoa2	3	1	4	4
Pessoa3	3	5	13	3
Pessoa4	3	9	14	10

Branch-and-Bound

Algoritmo Branch-and-Bound(x):

Entrada: uma instância x de um problema difícil de otimização (minimização)

Saída: uma solução ótima para x ou “sem solução” se nenhuma existir

$F \leftarrow \{(X,0)\}$

$b \leftarrow (+\infty, 0)$ //custo e configuração da melhor solução conhecida

Enquanto ! Vazio(F) **faça**

Retire de F uma configuração (x,y) mais promissora

Expanda(x,y), fazendo um pequeno conjunto de escolhas adicionais

Sejam (x1,y1), (x2,y2), ..., (xk,yk) o conjunto de novas configurações

Para cada nova configuração (xi, yi) **faça**

Verifique consistência de (xi, yi)

Se a verificação retorna “solução encontrada” **então**

Se custo c da solução para (xi,yi) for melhor do que b **então**

$b \leftarrow (c, (xi,yi))$

senão

descarte configuração (xi,yi)

Se a verificação retornar “sem saída” **então**

Descarte a configuração (xi,yi)

Senão

Se lb(xi,yi) é menor do que o custo de b **então**

$F \leftarrow F \cup \{(xi, yi)\}$

Senão

Descarte configuração (xi,yi) //o valor não interessa

FimEnquanto

Retorne b

lb é uma função de limite inferior ou superior

Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos –Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus
- TAMASSIA, ROBERTO; GOODRICH, MICHAEL T. (2004). Projeto de Algoritmos -Fundamentos, Análise e Exemplos da Internet
- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson