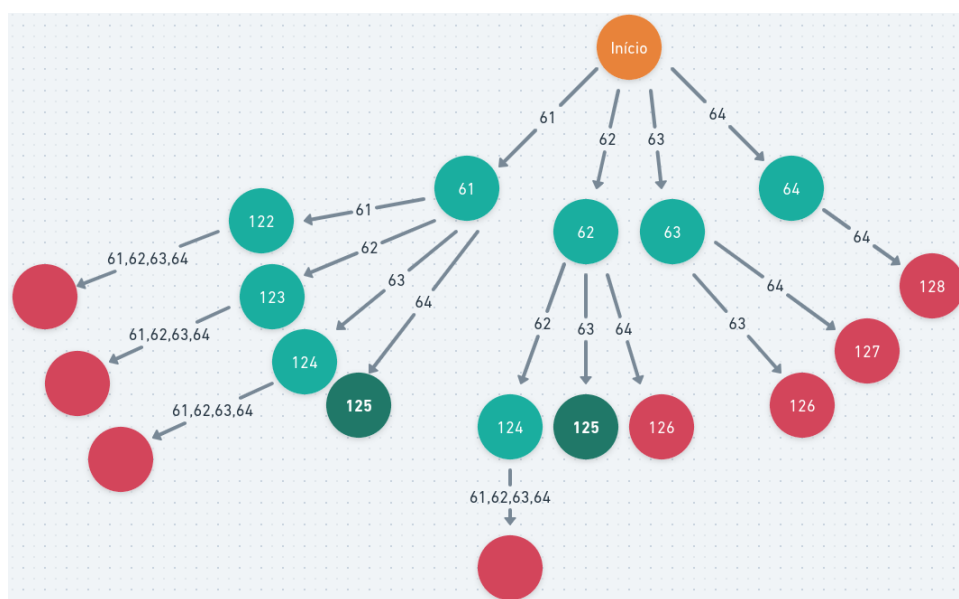


É possível sim obter soluções ótimas para o problema descrito utilizando o backtracking. Para isso, é necessário estabelecer apenas a seguinte restrição: se um conjunto de possibilidades ultrapassar o total T estabelecido, então esse caminho é descartado.

No entanto, é possível estabelecer outra restrição para otimizar a execução do algoritmo, que seria impossibilitar conjuntos de possibilidades iguais, mas com ordens diferentes, já que nesse problema a ordem dos elementos da solução é irrelevante. Para verificar essa restrição adicional, existem duas formas principais que são: armazenar os conjuntos que foram calculados anteriormente ou limitar a segunda geração da árvore para que não exista repetição.

O primeiro método é bem direto, basta armazenar os conjuntos anteriores e verificar se cada novo conjunto já pertence à lista de conjuntos antigos, se pertencer, este novo é descartado, já o segundo método pode ser feito através da seguinte lógica: cada elemento da primeira geração da árvore só irá gerar nós com ele mesmo e os outros elementos posteriores a ele, isso pode ser evidenciado no exemplo de árvore abaixo (que seguiu o exemplo da questão):



Como pode ser visto, o elemento 61 gera nós com ele mesmo e todos os outros, já o 62 gera com si próprio e todos os outros, exceto o 61, o 63 gera com ele mesmo e todos os outros, exceto o 61 e 62, e por fim o 64 gera com ele mesmo e todos os outros, exceto o 61, 62 e 63.

O backtracking é evidenciado nos nós vermelhos, que simbolizam quando uma combinação de possibilidades extrapola o valor limite, que nesse caso é 125. A restrição adicional também poderia ser feita através de backtracking, mas deixaria o algoritmo e a visualização da árvore mais complexos (além de reduzir o desempenho do algoritmo com verificações e iterações desnecessárias).