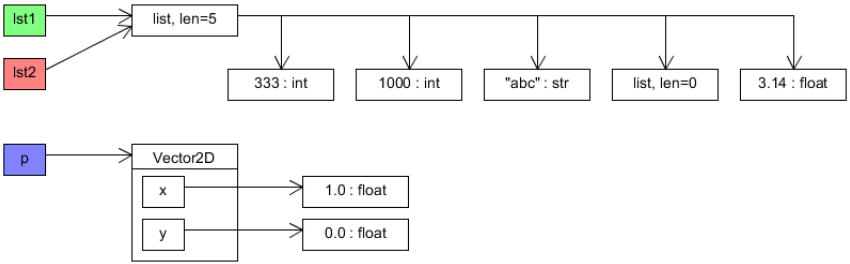
Głębokie kopie

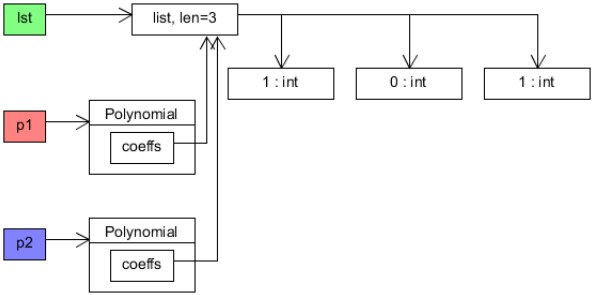
Diagram obiektów (uproszczony):

```
lst1 = [333, 1000, "abc", [], 3.14]
lst2 = lst1
p = Vector2D(x=1.0, y=0.0)
```



Głębokie kopie

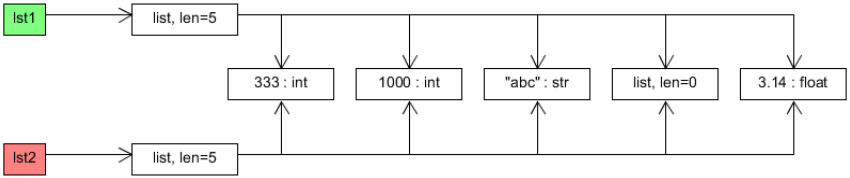
```
lst = [1, 0, 1]
p1 = Polynomial(lst)
p2 = Polynomial(lst)
lst[0] = ... # dotyczy wielu obiektow
```



Głębokie kopie

Kopia płytka: kopiuje obiekt, ale nie zawartość (np. obiekty nazywane przez atrybuty).

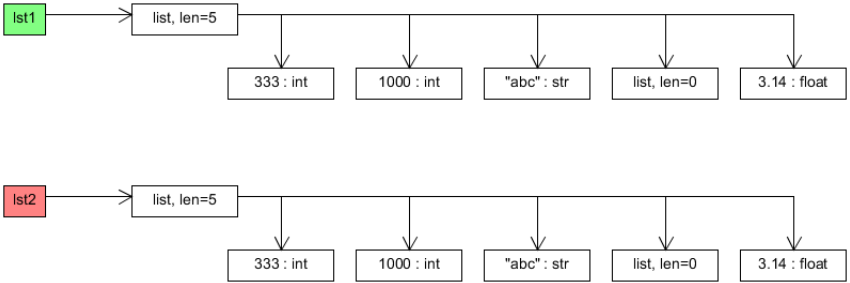
```
lst1 = [333, 1000, "abc", [], 3.14]
lst2 = lst1.copy() # lst1[:], list(lst1), copy.copy(lst1)
```



Głębokie kopie

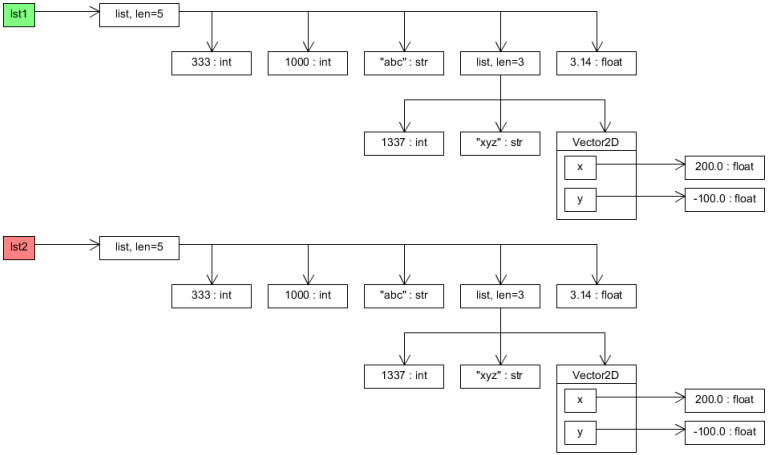
Kopia głęboka: rekurencyjnie kopiuje obiekt i wykonuje głębokie kopie zawartości.

```
lst1 = [333, 1000, "abc", [], 3.14]
lst2 = copy.deepcopy(lst1) # import copy
```



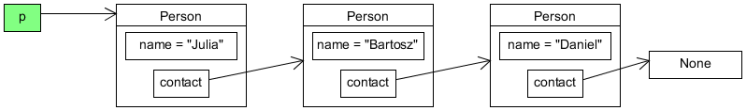
Głębokie kopie

Głębokie kopiowanie rekurencyjnie tworzy pełną kopię całej struktury powiązanej z obiektem.

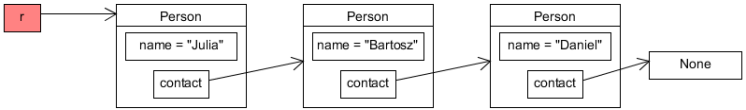


Głębokie kopie

Przykład: kontakt alarmowy.
Klasa Person zawierająca imię osoby (name) oraz atrybut contact oznaczający jej kontakt alarmowy (również instancję Person) lub None.

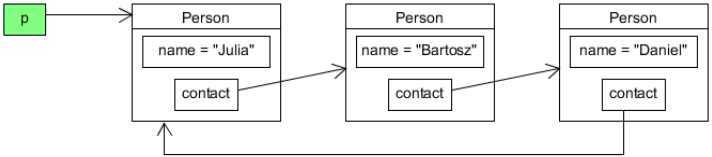



```
r = copy.deepcopy(p)
```

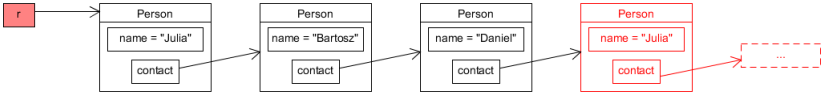


Głębokie kopie

Problem przy rekurencyjnym kopiowaniu: samoodniesienia.



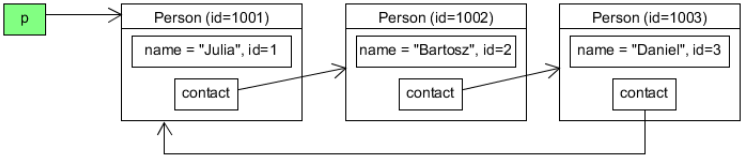

```
r = copy.deepcopy(p)
```



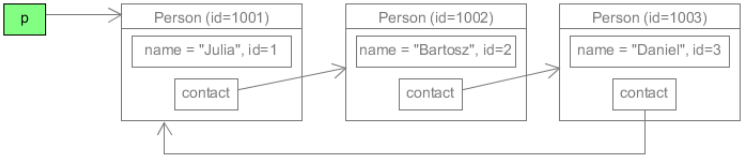
Głębokie kopie

Rozwiązanie: algorytm wykonujący głęboką kopię zapamiętuje identyfikatory skopiowanych już obiektów, stowarzyszając je z ich kopiami.

Przykładowe id obiektów z przykładu:

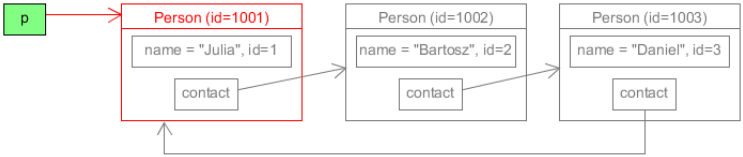


Głębokie kopie



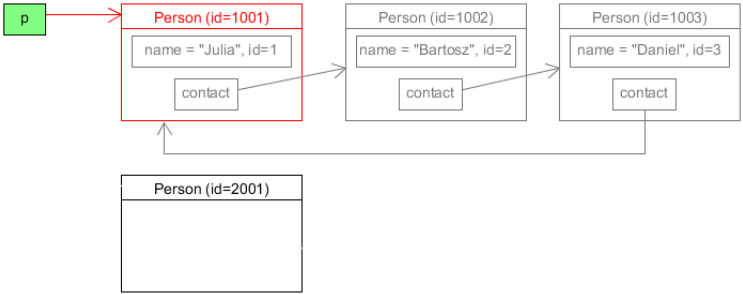
Skopiowane:

Głębokie kopie

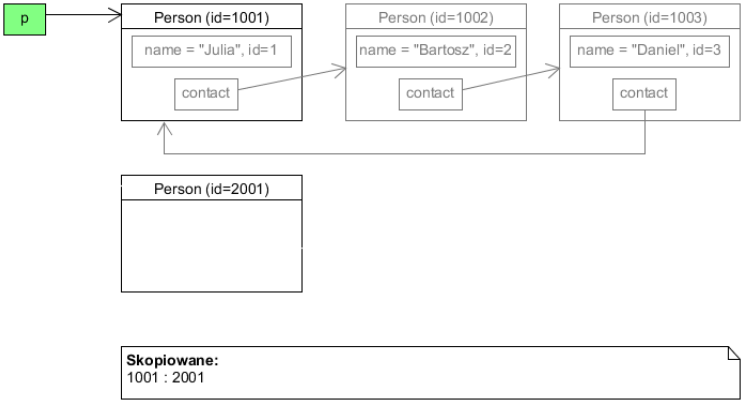


Skopiowane:

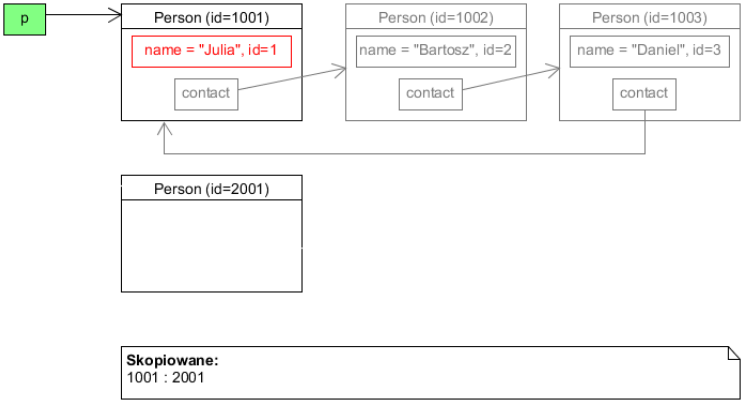
Głębokie kopie



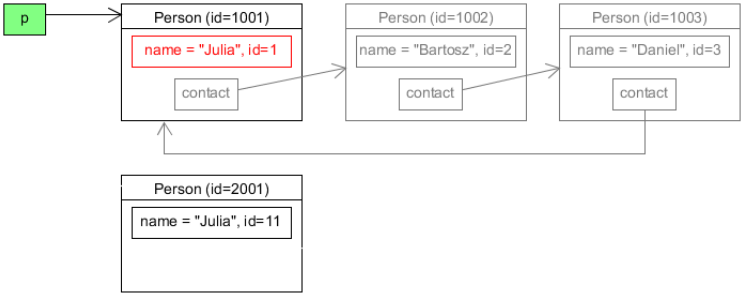
Głębokie kopie



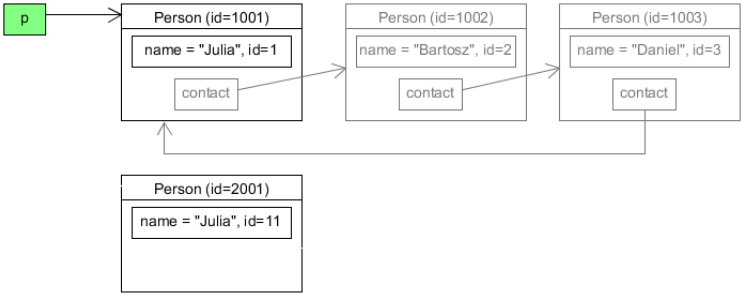
Głębokie kopie



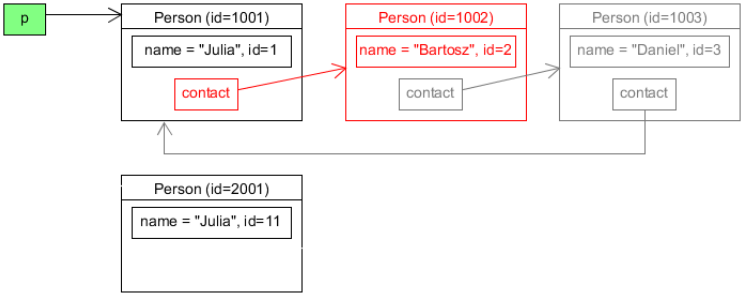
Głębokie kopie



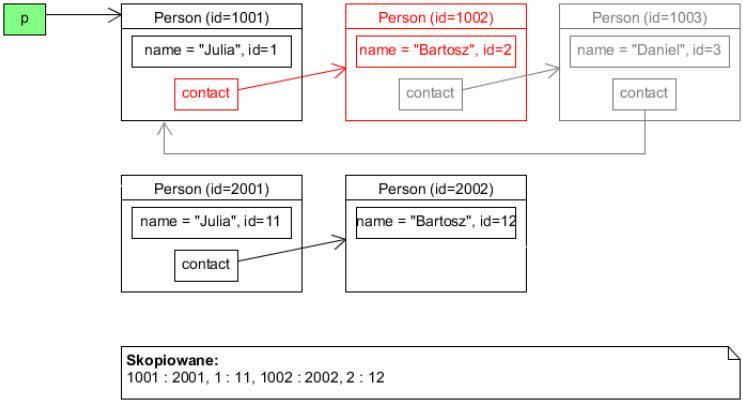
Głębokie kopie



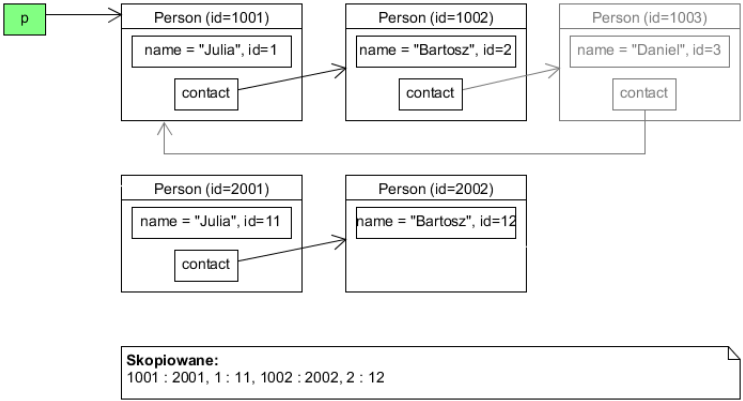
Głębokie kopie



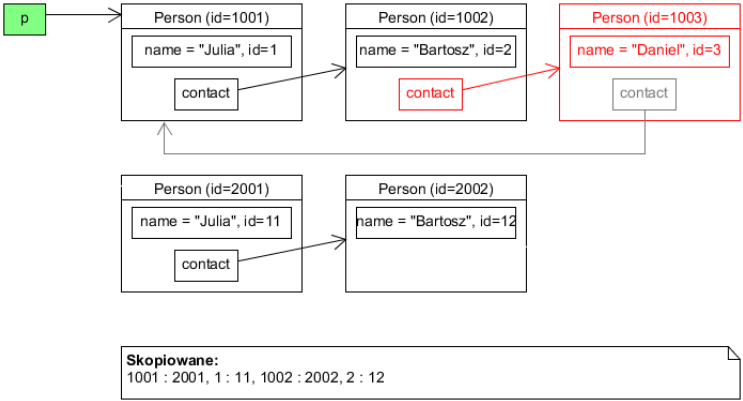
Głębokie kopie



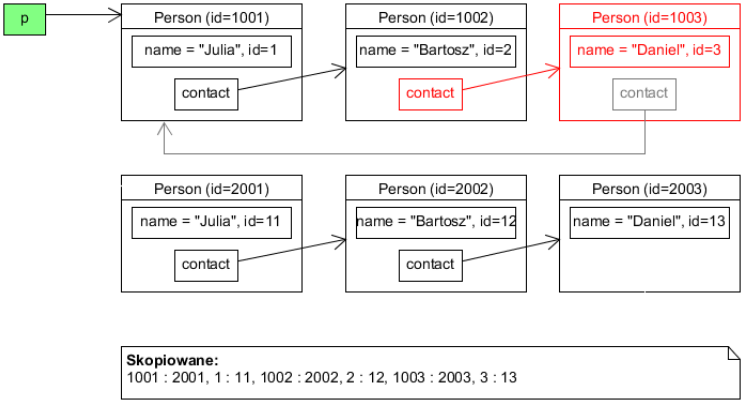
Głębokie kopie



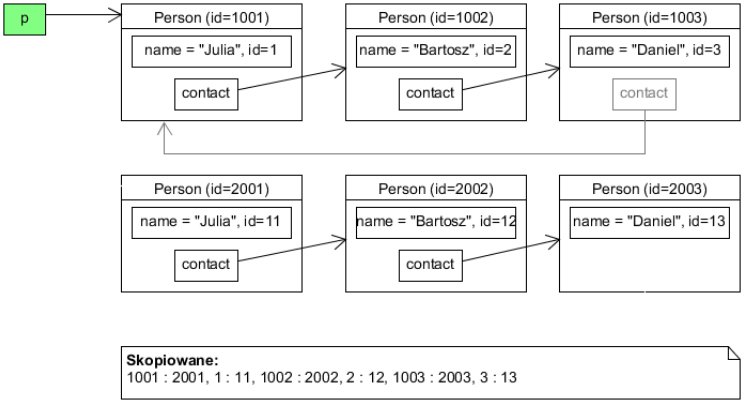
Głębokie kopie



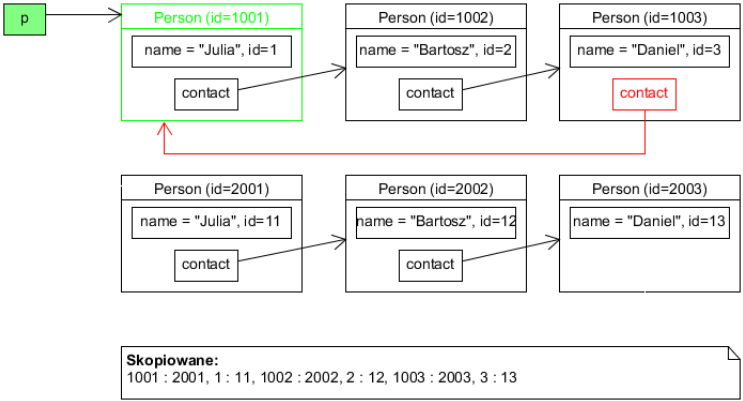
Głębokie kopie



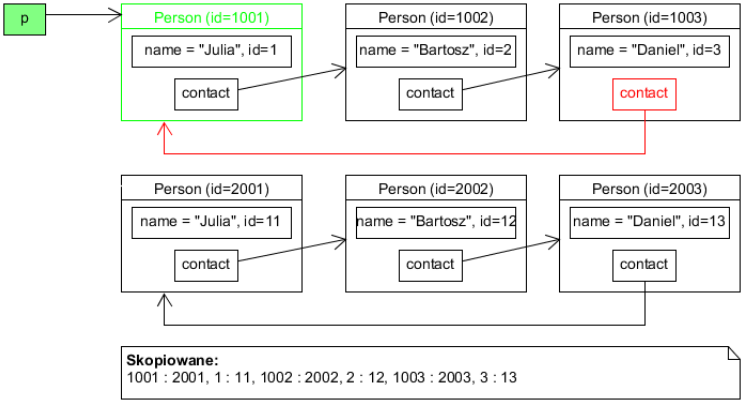
Głębokie kopie



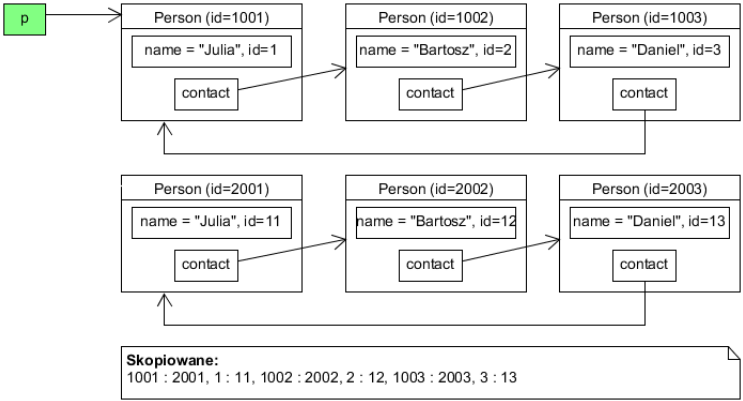
Głębokie kopie



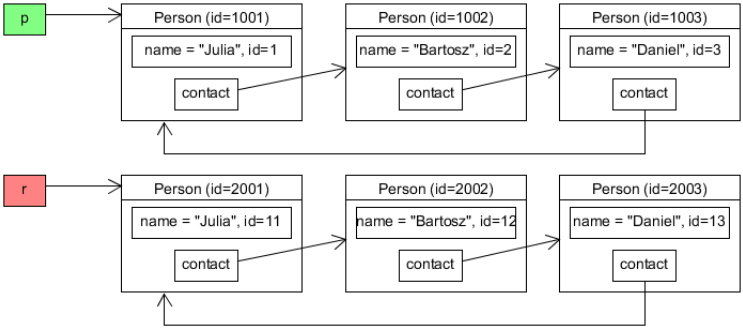
Głębokie kopie



Głębokie kopie

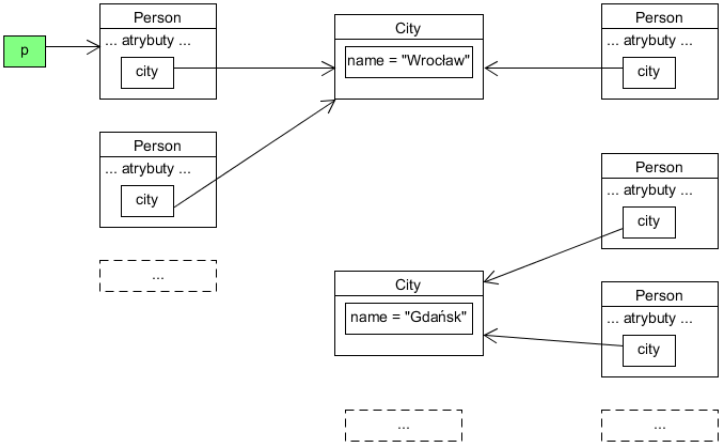


Głębokie kopie



Głębokie kopie

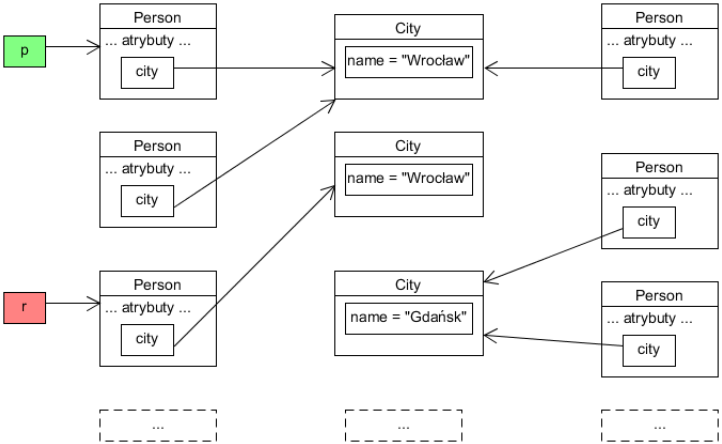
A co, jeśli nie chcemy pełnej kopii struktury?
Przykład:



Głębokie kopie

A co, jeśli nie chcemy pełnej kopii struktury?

```
r = copy.deepcopy(p)
```



Głębokie kopie

Specjalna metoda __deepcopy__ pozwala na zdefiniowanie sposobu, w jak copy.deepcopy ma wykonać głęboką kopię obiektu.

```
    def __deepcopy__(self, memo):
        # ...
```

memo - słownik pamiętający skopiowane już obiekty (w parach (id, obiekt-kopia)).
Metoda zwraca głęboką kopię obiektu.
Przykładowe rozwiązanie* problemu: zdefiniowanie __deepcopy__ dla klasy City tak, żeby zwracała oryginał i nie tworzyła kopii.

```
class City:
    # ...
    def __deepcopy__(self, memo):
        return self
```

* - niekoniecznie optymalne dla tego problemu