

1. Recursion

Example 1: Max Couple at Inverted Indexes



Recursion1

```
1 def max_couple(list, length):
2     return max_couple_helper(list, length)
3
4 def max_couple_helper(list, length, max = 0, start = 0):
5     if length == 0 or (length == 1 and list[length + start - 1] <= max):
6         return max
7
8     if length == 1 and list[length + start - 1] > max:
9         return list[length + start - 1]
10
11    if list[start] + list[length + start - 1] > max:
12        max = list[start] + list[length + start - 1]
13
14    return max_couple_helper(list, length - 2, max, start + 1)
```

Example 2: Twin Neighbors



Recursion2

```
1 def twin_neighbours(my_list):
2     return twin_helper(my_list, len(my_list) - 1)
3
4 # Helper for q1
5 def twin_helper(my_list, last_index, count=0):
6     if last_index == 0:
7         return 0
8     if my_list[last_index] == my_list[last_index - 1]:
9         return 1 + twin_helper(my_list, last_index - 1)
10    else:
11        return twin_helper(my_list, last_index - 1)
```

Example 3: יעני פיבונצ'י



Recursion3

```
1 def like_fibo(nth):
2     if nth <= 3:
3         return nth
4
5     elif nth % 2 == 0:
6         return like_fibo(nth - 1) + like_fibo(nth - 2) + like_fibo(nth - 3)
7     else:
8         return abs(like_fibo(nth - 1) - like_fibo(nth - 3))
```

Example 4: מספר מתחלף



Recursion4

```
1 def is_switched_number(number):
2     if number < 10:
3         return True
4     dig0 = number % 10
5     dig1 = number // 10 % 10
6     if (dig0 % 2 == 0 and dig1 % 2 == 0) or (dig0 % 2 != 0 and dig1 % 2 != 0):
7         return False
8
9     return is_switched_number(number // 10)
10 '''
```