

BruteForceSearch

August 29, 2025

0.1 Brute Force Search, short version

```
[4]: def bfs_aux(a, b):  
    if not b:  
        return [] if True in [a[i] > a[i + 1] for i in range(len(a) - 1)] else a  
  
    for i in range(len(b)):  
        s = bfs_aux(a + [b[i]], b[:i] + b[i + 1:])  
        if s:  
            return s  
  
    return []  
  
def bfsearch(a):  
    return bfs_aux([], a)
```

```
[5]: bfsearch([5, 6, 5, 1, 2, 1, 3])
```

```
[5]: [1, 1, 2, 3, 5, 5, 6]
```

0.2 Brute Force Search, generator version

```
[6]: # True if input array is sorted (not optimal ...).  
def is_sorted(a):  
    return not (True in [a[i] > a[i + 1] for i in range(len(a) - 1)])  
  
# Generator function for permutations.  
def gen_perms(a, b):  
    if not b:  
        yield a  
  
    for i in range(len(b)):  
        yield from gen_perms(a + [b[i]], b[:i] + b[i + 1:])  
  
# Brute force search function.  
def bfsearch(a):  
    for e in gen_perms([], a):  
        if is_sorted(e):
```

```
return e
```

```
[7]: bfsearch([5, 6, 5, 1, 2, 1, 3])
```

```
[7]: [1, 1, 2, 3, 5, 5, 6]
```

```
[ ]:
```