tf.contrib.framework.nest.pack_sequence_as

```
pack_sequence_as(
    structure,
    flat_sequence
)
```

Defined in `tensorflow/python/util/nest.py`.

Returns a given flattened sequence packed into a given structure.

If `structure` is a scalar, `flat_sequence` must be a single-element list; in this case the return value is `flat_sequence[0]`.

If `structure` is or contains a dict instance, the keys will be sorted to pack the flat sequence in deterministic order. This is true also for `OrderedDict` instances: their sequence order is ignored, the sorting order of keys is used instead. The same convention is followed in `pack_sequence_as`. This correctly repacks dicts and `OrderedDict`s after they have been flattened, and also allows flattening an `OrderedDict` and then repacking it back using a correponding plain dict, or vice-versa. Dictionaries with non-sortable keys cannot be flattened.

Args:

- `structure`: Nested structure, whose structure is given by nested lists, tuples, and dicts. Note: numpy arrays and strings are considered scalars.
- `flat_sequence`: flat sequence to pack.

Returns:

- `packed`: `flat_sequence` converted to have the same recursive structure as `structure`.

Raises:

- `ValueError`: If `flat_sequence` and `structure` have different element counts.
- `TypeError`: `structure` is or contains a dict with non-sortable keys.

Last updated November 2, 2017.