# heapp

*Heap protocol.*

author:
`Richard O'Keefe; adapted to Logtalk by Paulo Moura and Victor Lagerkvist.`
version:
`1.01`
date:
`2010/11/13`

compilation:
`static, context_switching_calls`

*(no dependencies on other files)*

## Public interface

### insert/4

*Inserts the new Key-Value pair into a heap, returning the updated heap.*

compilation:
`static`

template:
`insert(Key,Value,Heap,NewHeap)`

mode - number of solutions:
`insert(+key,+value,+heap,-heap) - one`

### insert_all/3

*Inserts a list of Key-Value pairs into a heap, returning the updated heap.*

compilation:
`static`

template:
`insert_all(List,Heap,NewHeap)`

mode - number of solutions:
`insert_all(@list(pairs),+heap,-heap) - one`

### delete/4

*Deletes and returns the top Key-Value pair in OldHeap and the resulting NewHeap.*

compilation:
`static`

template:
`delete(Heap,Key,Value,NewHeap)`

mode - number of solutions:
`delete(+heap,?key,?value,-heap) - zero_or_one`

### merge/3

*Merges two heaps.*

compilation:
`static`

template:
```
merge(Heap1,Heap2,NewHeap)
```
mode - number of solutions:
```
merge(+heap,+heap,-heap) - one
```

## empty/1

*True if the heap is empty.*

compilation:
```
static
```
template:
```
empty(Heap)
```
mode - number of solutions:
```
empty(@heap) - zero_or_one
```

## size/2

*Returns the number of heap elements.*

compilation:
```
static
```
template:
```
size(Heap,Size)
```
mode - number of solutions:
```
size(+heap,?integer) - zero_or_one
```

## as_list/2

*Returns the current set of Key-Value pairs in the Heap as a List, sorted into ascending order of Keys.*

compilation:
```
static
```
template:
```
as_list(Heap,List)
```
mode - number of solutions:
```
as_list(+heap,-list) - one
```

## as_heap/2

*Constructs a Heap from a list of Key-Value pairs.*

compilation:
```
static
```
template:
```
as_heap(List,Heap)
```
mode - number of solutions:
```
as_heap(+list,-heap) - one
```

## top/3

*Returns the top Key-Value pair in Heap. Fails if the heap is empty.*

compilation:
```
static
```
template:
```
top(Heap,Key,Value)
```

mode - number of solutions:

```
top(+heap,?key,?value) - zero_or_one
```

### top_next/5

*Returns the top pair, Key1-Value1, and the next pair, Key2-Value2, in Heap. Fails if the heap does not have at least two elements.*

compilation:

```
static
```

template:

```
top_next(Heap,Key1,Value1,Key2,Value2)
```

mode - number of solutions:

```
top_next(+heap,?key,?value,?key,?value) - zero_or_one
```

## Protected interface

*(none)*

## Private predicates

*(none)*