Package: binpackr (via r-universe)

September 5, 2024

Title Fast 1d Bin Packing

Version 0.1.1.9000

Description Implements the First Fit Decreasing algorithm to achieve one dimensional heuristic bin packing. Runtime is of order O(n log(n)) where n is the number of items to pack. See ``The Art of Computer Programming Vol. 1'' by Donald E. Knuth (1997, ISBN: 0201896834) for more details.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

LinkingTo cpp11

Suggests testthat (>= 3.0.0), hedgehog (>= 0.1), BBmisc (>= 1.13)

Config/testthat/edition 3

URL `https://github.com/lschneiderbauer/binpackr`, `https://lschneiderbauer.github.io/binpackr/`

BugReports `https://github.com/lschneiderbauer/binpackr/issues`

Repository https://lschneiderbauer.r-universe.dev

RemoteUrl https://github.com/lschneiderbauer/binpackr

RemoteRef HEAD

RemoteSha 00cc6d8e8c0e6df21bd4e8610971de3be0d327e4

Contents

bin_pack_ffd 2

Index **3**

`bin_pack_ffd` *1D bin packing "First Fit (Decreasing)" algorithm*

Description

1D bin packing "First Fit (Decreasing)" algorithm

Usage

```
bin_pack_ffd(x, cap, sort = TRUE)
```

Arguments

`x` A numeric vector of item sizes to be fit into bins. Each value represents the size of an atomic item. If a value is `NA` it is ignored and the corresponding result will also be `NA`.

`cap` Bin capacity in units of values in `x`. A scalar value. If an individual item size is above `cap` a single bin is reserved for this item.

`sort` Determines whether the input vector should be sorted in decreasing order before applying the "First Fit" algorithm ("First Fit Decreasing").

Details

See Wikipedia for a concise introduction or "The Art of Computer Programming Vol. 1" by Donald E. Knuth (1997, ISBN: 0201896834) for more details.

Value

A integer vector of labels of the same length as `x`. The integer label at position `i` determines the assignment of the `i`th item with size `x[i]` to a bin. If the value `x[i]` is `NA` the result label at position `i` will also be `NA`.

Examples

```
# Generate a vector of item sizes
x <- sample(100, 1000, replace = TRUE)

# Pack those items into bins of capacity 130
bins <- bin_pack_ffd(x, cap = 130)

# Number of bins needed to pack the items
print(length(unique(bins)))
```

Index

bin_pack_ffd, 2