/\*\*

\* The default initial capacity - MUST be a power of two.

\*/

**static final int DEFAULT\_INITIAL\_CAPACITY = 1 << 4; // aka 16**

/\*\*

\* The maximum capacity, used if a higher value is implicitly specified

\* by either of the constructors with arguments.

\* MUST be a power of two <= 1<<30.

\*/

**static final int MAXIMUM\_CAPACITY = 1 << 30;**

/\*\*

\* The load factor used when none specified in constructor.

\*/

**static final float DEFAULT\_LOAD\_FACTOR = 0.75f;**

**/\*\***

**\* The bin count threshold for using a tree rather than list for a**

**\* bin. Bins are converted to trees when adding an element to a**

**\* bin with at least this many nodes. The value must be greater**

**\* than 2 and should be at least 8 to mesh with assumptions in**

**\* tree removal about conversion back to plain bins upon**

**\* shrinkage.**

**\*/**

**static final int TREEIFY\_THRESHOLD = 8;**

**/\*\***

**\* The bin count threshold for untreeifying a (split) bin during a**

**\* resize operation. Should be less than TREEIFY\_THRESHOLD, and at**

**\* most 6 to mesh with shrinkage detection under removal.**

**\*/**

**static final int UNTREEIFY\_THRESHOLD = 6;**

**/\*\***

**\* The smallest table capacity for which bins may be treeified.**

**\* (Otherwise the table is resized if too many nodes in a bin.)**

**\* Should be at least 4 \* TREEIFY\_THRESHOLD to avoid conflicts**

**\* between resizing and treeification thresholds.**

**\*/**

**static final int MIN\_TREEIFY\_CAPACITY = 64;**