

Forum:[Hack Forums](#)

Thread: [In what situations are Bags more useful to use compared to other collection types?](#)

SUI-3183372 - October 18, 2017, 12:45 pm

I have to write a paper on the uses and implementation of the collection type: bags.

I'm looking for some more insight from people so I can present more examples.

In what situations would you prefer to use bags compared to other collection types?

randomized patterns - October 18, 2017, 3:13 pm

You haven't told us what a bag is (which is obviously not a traditional data structure in java). Describe the data structure and or post sample code of a "Bag."

SUI-3183372 - October 18, 2017, 5:42 pm

<http://algs4.cs.princeton.edu/13stacks/>

<https://commons.apache.org/proper/commons-collections/javadocs/api-2.1.1/org/apache/commons/collections/Bag.html>

<https://stackoverflow.com/questions/28442310/data-structures-creating-a-bag-in-java>

<https://stackoverflow.com/questions/14651105/building-a-bag-class-in-java?rq=1>

Bags are very similar to sets, bags can just contain duplicates.

A <bag> is an unordered collection, which can contain duplicated elements. That means if you persist a bag with some order of elements, you cannot expect the same order retains when the collection is retrieved. There is not a "bag" concept in Java collections framework, so we just use a java.util.List corresponds to a <bag>.

A <set> is similar to <bag> except that it can only store unique objects. That means no duplicate elements can be contained in a set. When you add the same element to a set for second time, it will replace the old one. A set is unordered by default but we can ask it to be sorted. The corresponding type of a in Java is java.util.Set.

GammaG - October 19, 2017, 12:45 am

That means it has an unordered list of maybe doubled or maybe not doubled elements of which you never can be sure which one you got in the end?

Mh ok someone has some usefull case for this? Can't think of one

SUI-3183372 - October 19, 2017, 1:13 am

(Oct 19, 2017 12:45 AM)GammaG Wrote: [\[CITE\]](#) That means it has an unordered list of maybe doubled or maybe not doubled elements of which you never can be sure which one you got in the end?

Mh ok someone has some usefull case for this? Can't think of one

It is very easy to search through bags. Searching a bag will give us $O(\log n)$ or $O(1)$ searches. A linked list, for example, is going to give $O(n)$ searches.

Quote:In general, the Set or Bag interfaces will be more suitable when you often need to perform a search or delete operations. If you just need to add to the end of the collection and iterate over it,

there wouldn't be much difference.

GammaG - October 19, 2017, 5:48 am

(Oct 19, 2017 01:13 AM)Previously Known As.... Wrote: [\[CITE\]](#) It is very easy to search through bags. Searching a bag will give us $O(\log n)$ or $O(1)$ searches. A linked list, for example, is going to give $O(n)$ searches. Ah good to know so it is usefull if you know that you dont have doubled elements anyway.

randomized patterns - October 20, 2017, 1:34 pm

Ehh I'd like to do a write up but don't have the time right now.

Look into practical uses and compare to other data structures.

??Austin?? - October 21, 2017, 2:42 am

Bags are a datastructure that allow you to: `add(a)`, `remove(a)`, `iterate()`, `size()`, and `contains(a)`. They may contain duplicates and have no defined order or obligation to order.

I see them as a semantic.

This can be useful to emphasize that order does not matter when performing operations on your data. Examples (spellchecker words vs.. dictionary), mean, std dev, summation, etc. They can also be used to emphasize that the data cannot be meaningfully sorted. Examples might include vertices of an object in space(or positions of players in a game), data with no resolved meaning(such as input to a neural network, images/media with no meta-data, components on the screen), etc. There is also the case in which order can matter but control over said order should not be controled by the user of the bag. This could be useful if the list must always be sorted.

It should also be noted a bag can be implemented using arrays, dynamic arrays, linked lists, etc. It really doesn't matter.

http://web.engr.oregonstate.edu/~sinisa/courses/OSU/CS261/CS261_Textbook/Chapter08.pdf
<https://algs4.cs.princeton.edu/13stacks/>