

Symbol Tables

A. For each application, choose the best types for the ST:

1. For each calendar year since 1748, the number of students who graduated from Princeton that year.	d	a. ST<String, Integer>
2. Dictionary of all synonyms of all English words.	e	b. ST<String, Boolean>
3. Count frequency of words in a book.	a	c. ST<Integer, String>
4. List people living at every postal address and their ages.	f	d. ST<Integer, Integer>
5. Track how each senator voted (yes or no) on a motion.	b	e. ST<String, String[]>
6. Look up the state containing any given zip code.	c	f. ST<String, ST<String, Integer>>

B.

1. Say we have a program that uses a double[]. Can we replace it with ST<Integer, Double> in all cases? Why or why not?

Yes, you can replace all element access with .get() and assignment with .put().

2. Now, vice versa. Say we have a program that uses ST<Integer, Double>. Can we replace it with a double[] in all cases? Why or why not?

No, you can do things with STs that you can't do with arrays. In an ST, you can have negative indices or skip indices. You can also change the size easily/quickly.

3. What are the benefits of using an ST?

STs are more flexible than arrays and you don't need to know the size in advance.

4. What are the benefits of using an array?

Arrays are faster and take up less space.