

## Part B Questions

- 1 Which version of the tree had the fastest processing (Section A- built from sorted ascending, Section B- build from shuffled, or Section C- built from sorted descending)? Why do you think that is?

Section B- build from shuffled is the fastest.

Section A: Time required to build tree: 1398 milliseconds

Section B: Time required to build tree: 84 milliseconds

Section C: Time required to build tree: 24073 milliseconds

Section B is the smallest, so it's the fastest.

- 2 How can you explain the difference in building and processing time between the two versions of the tree that were built from the sorted lists Sections A and C)? The tree built from the ascending list was faster. Why?
  - Hint: Try drawing a small tree to see what is going on. Perhaps a tree built from 1, 1, 2, 2, 3, 4, 4, 5 and from 5, 4, 4, 3, 2, 2, 1.

The tree built from ascending list has shorter height; while the one built from descending list is like a linked chain, has the same height as its nodes number.

- 3 In Section B, which processing was faster- the tree built from the shuffled list or the list? How would you describe the big-o of what was going on in the processing with these two structures?

The tree built from the shuffled list is faster.

When building the tree, reading report input is like *add* method.

So add node is  $O(\log n)$ , and add to List is  $O(n)$ .

- 4 What characteristic of a tree affects its efficiency?  
The sorted status and height of a tree affects its efficiency.