Question - 1

Sorting

SCORE: **5 points**

Choose two sorting algorithms that have similar implementation logic.

- [x] Insertion
- [x] Shell
- [ ] Merge
- [ ] Selection

Question - 2

Sorting

SCORE: **5 points**

Based on the difference between java primitives and object references, *Arrays.sort()* uses a different sorting algorithm depending on the underlying type. Based on your knowledge of the behavior of various sorting methods, specifically which algorithms are chosen for primitives and references?

- [ ] Insertion, Merge
- [ ] Merge, Shell
- [x] Quick, Merge
- [ ] Quick, Shell

Question - 3

Sorting

SCORE: **5 points**

Choose all stable algorithms.

- [ ] Selection
- [x] Insertion
- [ ] Quick
- [ ] Heap

Question - 4

SCORE: **5 points**

Sorting

Given an array of identical elements, what algorithms perform similarly in time complexity?

- [x] Selection
- [x] Quick
- [ ] Insertion
- [ ] Shell

Question - 5

SCORE: **5 points**

Easy blank

Problem Statement

Given definitions as following:

A full binary tree(sometimes proper binary tree or 2-tree) is a tree in which every node other than the leaves has two children.

A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.

According to definitions, a <blank 1> is a <blank 2> by deafult.

Answers

<blank 1> : [full binary tree]
<blank 2> : [complete binary tree]

Question - 6

SCORE: **25 points**

Schedule

We all have many things to do each day. Hang out with friends, play video games, chat with our GF or BF, buy a new pair of Air Jordan shoes, feed our pets and so on (almost everything except study :P . We keep telling ourselves we need to study every night though).
All these things have a different level of importance and we decide to represent it by an integer. A high integer means more importance and we want to finish them **first**.
Moreover, we are single threaded so we could only focus on one thing and **until we finish it then could we start another one**.
We may come up with activities when we are busy so we put it **into our to-do-list properly**. But if we are free at that time, we get down to work on the next activity **immediately**.
If we are free and there is nothing to do, we **wait** for activities.

Given an array of activities in the form of [String name, int priority, int start time, int duration], return the finishing order of them.
Such arrays are **wrapped** by a class, which is used in test cases.

Example:

GIven { [buy shoes,5,2,30], [wash clothes,6,40,50], [do homework,10,10,100] } ->

{ buy shoes, do homework, wash clothes}