Notice

- Follow Java Code Conventions; otherwise you cannot get full points. The requirement on naming (section 4) and indentation (section 9) are hard. Other parts are suggestive.
- Make sure your codes can be compiled by command line. If you've problems, this and this web pages may help, and you're welcome to ask the graders.
- For each question, you submission should contain a .java source file and a screenshot of its execution. If you don't know how to take a screenshot, this website may help.
- Submit on learn@polyu, no hard-copies are required.

Mandate questions

1. (25 points) Finish `CourseRegistrySorted.java` and `CourseRegistryUnsorted.java` given in Lab 3. You need to implement all the methods: `register, drop, search`.
2. (25 points) Finish `BinarySort.java`. You are asked to sort the students according to their gender, which can be either 0 (female) or 1 (male), such that all female students come before male students. Your algorithm must run in linear-time $O(n)$, and you are *not* allowed to use another array. The extra storage used by your algorithm, besides the given array, should be no more than constant size (this is also called *in place* sort).
3. (25 points) Write a method to recursively remove all adjacent duplicates from a given string, e.g., `"careermonk"` $\rightarrow$ `"camonk"` (note that two `r`'s were not adjacent initially) and `"mississippi"` $\rightarrow$ `"m"`.
   Hint: Similar as the multi-parentheses matching.
4. (25 points) Try to improve any method I wrote in the class. You need to propose at least three improvement ideas and try to test your ideas. Record your test results: running time. It's perfectly OK that your improvement idea turns out to be worse than my version.
   Hint: You need to use a big number of random data to make the difference discernible. You may refer to the lab 2 code for how to do this and how to record running time.

Challenging question

You're suggested to try this to get bonus points and better understanding.

1. Given an array of integers, you are asked to give an algorithm to find the second smallest of them with at most $n+\lceil \log n \rceil - 2$ comparisons. Note that this requirement is stronger than linear-time: We're not using the Big-Oh notation, so $2n$ comparisons wouldn't count.

   Hint: divide and conquer to find the smallest; where is the second smallest?