IT 427, Design and Analysis of Algorithms

Programming Assignment 5: RNA Folding

Due date: Oct. 30, 2023, Monday, 11:55 PM 40 points (30 on programs, 10 on report)

For this assignment, you will use dynamic programming technique to find the optimal secondary structure of a given RNA, i.e., maximize the number of basic base pairs (i.e, A-U and C-G pairs, with some constraints such as, no crossover pairs on the strand, no sharp folding, 1-1 match. Check Chapter 6.5 of the textbook for detailed description.) This problem is known as the RNA folding problem.

As in the previous programming assignment, prepare your programs on our Linux Server.

- Make a directory `asg5` under your `IT427`, i.e., ∼`/IT427/asg5/`. All needed files for this assignment should be saved under your ∼`/IT427/asg5` before run `submit427.sh`.
- Check the contents of my `/home/ad.ilstu.edu/cli2/Public/IT427/asg5` and copy all files to your own ∼`/IT427/asg5`.

Program requirement: The name of the program should be `RNAfolding.java`. Code all needed classes and methods in one program file, i.e., do not create any external class for this project. I will compile and run your program on our Linux server as follows, where `RNA1.txt` is the input file. If your program fails to compile, you will get 0 point. I may test your program on a different RNA file.

```
javac RNAfolding.java
java RNAfolding RNA1.txt
```

Input: The input text file contains several single-stranded RNA molecules $b_1b_2 \dots b_n$ as follows (Note that, the index of the RNA sequence starts from 1:

```
**RNA-1:
UCUUCCUACAGACGAAACGUAAGCCUGCGCCCAUAGUUCCAUUUAUUCGA

**RNA-2:
AAUUGCCCACAAUUAAUUGGCGCAAUUGCCGAGGGAACGAACGUUCCCUC

**RNA-3:
UAUGUGUAAGUGAACGCGUCGUACGACGCUUCUGCUCCGUGUUCGCUGUC

......
......
```

There are 3 RNA files in the public directory for your testing. I may use another files to test your program.

Output: The program should list the required information formatted as follows: For each RNA sequence, the first line prints the RNA number and the length of the sequence followed by a list of base pairs in an optimal folding, one line for each pair. At the end of the list, print the total number of base pairs. The last line should print your name to declare this is your work.

```
Folding RNA in file RNA1.txt

** RNA-1, length=50, Optimal secondary structure:
U-A (1,12)
C-G (2,11)
U-A (3,8)
C-G (13,49)
G-C (14,48)
A-U (15,44)
A-U (16,43)
A-U (17,42)
G-C (19,39)
A-U (21,38)
A-U (22,37)
C-G (24,36)
U-A (26,33)
G-C (27,32)
Total number of base pairs:14

** RNA-2, length=50, Optimal secondary structure:
A-U (1,27)
......
......
C-G (38,43)
Total number of base pairs:18

* RNA-3, length=50, Optimal secondary structure:
......
......

All by Chung-Chih Li
```

Submission: Programs (30 points) and Reports (10 points)

Submission details are same as the previous assignment. Run the submission script with the submission number changed to 5, but you can use the same secret name as follow:

```
bash /home/ad.ilstu.edu/cli2/Public/IT427/submit427.sh peekapoo 5
```

Note that, since I will keep updating `submit427.sh` for different assignment, you have to run the script from my `/home/ad.ilstu.edu/cli2/Public/IT427/` directly for the most recent updated version, i.e., don't copy it to your own directory. Also, once you run `submit427.sh`, don't make any change on the target directory, including compile and run your programs. That will mess up the permission. You will lose significant points if you fail to follow the instructions.

★ Any plagiarism will receive 0 and be reported to the school ★

1. Programs: 30 points. Submission on Linux server.

The score is based on the correctness and the programming style, which includes efficiency, appropriateness of data structures, and documentation of your programs. At the beginning of every program file, put a section of comments including (1) your full name, (2) student ID, (3) a pledge of honesty that you do not copy/modify from other's codes and (4) a declaration of your copyright that no one else should copy/modify the codes. You will receive:

(a) 30 points: No error with a good programming style.
(b) 25 $\sim$ 29 points: Minor error and fair programming style.
(c) 20 $\sim$ 24 points: Some error and not so good but acceptable programming style.
(d) 10 $\sim$ 19 points: Too many error and bad programming style, but meaningful.
(e) 5 $\sim$ 9 points: Compilable but not working and the program must show reasonable trying.
(f) 0 points : Fail to meet any of aforementioned qualities or plagiarism involved.

2. Report: 10 points. Submission through Canvas.

You have to write up a report and prepare it in pdf format. You don't have to put program output on the report as I will run and exam your program directly on some different input files.

(a) 3 points on the cover page, and

(b) 7 points on algorithm descriptions, summary of the methods, data structures, and efficiency analysis on time and space in details in terms of big-O notations. If there is any difficulties encountered in this assignment, you can report it.

If your analysis is not clearly related to your program and provides sufficient justification, your report score will not be higher than 5/10.