

CS313 Exercise 9 Cover Page

Name(s): _____

In the *TIME* column, please estimate the time you spent on the parts of this exercise. Please try to be as accurate as possible; this information will help us to design future exercises.

PART	TIME	SCORE
Exercise		

Task 1: RNA Structure with Maximum Number of Basepairs

For the RNA sequence ACAAAGUGGAUACC, complete the table below to determine the secondary structure that has the maximum number of basepairs. You should report both (a) the maximum number of basepairs that can occur in a secondary structure of the RNA and (b) the structure containing the maximum number of basepairs.

C														0
C													0	0
A												0	0	0
U											0	0	0	0
A										0	0	0	0	0
G									0	0	0	0	1	1
G								0	0	0	0	0	1	2
U							0	0	0	1	1	1	1	2
G						0	0	0	0	1	1	1	2	2
A					0	0	1	1						
A				0	0	0	1	1						
A			0	0	0	0	1	1						
C		0	0	0	0	1	1	2						
A	0	0	0	0	0	1	2	2						
	A	C	A	A	A	G	U	G	G	A	U	A	C	C

Task 2: Computational Identification of RNA Structure with Maximum Basepairings

Download the `RNA_Structure` folder from the download folder on the CS server. In this task, you will be working with the `RNA_Basepair` application. Compile and execute the `RNA_Basepair` application. The application requires one command line argument corresponding to a FASTA formatted file containing an RNA sequence. You can find some example files containing RNA sequences in the `sequences` sub-folder that you downloaded.

Given an RNA sequence, the `RNA_Basepair` application computes the secondary structure that has the maximum number of basepairings. In the program's output, note the parenthesis-dot notation above the RNA sequence. Parenthesis-dot notation is used to indicate a secondary structure. A dot (i.e., a period) indicates the nucleotide does not pair in the structure. An open parenthesis indicates the nucleotide does pair with the corresponding closed parenthesis nucleotide in the structure.

Unfortunately, the `RNA_Basepair` application is incomplete. When searching for the structure with the maximum number of basepairings, it does not consider structures with bifurcations. Thus, if the structure with the maximum basepairings contains a bifurcation, the `RNA_Basepair` application will not identify the structure but rather report a suboptimal structure.

For the provided tRNA sequence, how many basepairs does the structure (*excluding* bifurcations) with maximum basepairings contain?

For the provided MiR-96 microRNA precursor sequence, how many basepairs does the structure (*excluding* bifurcations) with maximum basepairings contain?

Your job is to modify the `computeStructureWithMaximumBasepairs` method in the `RNA_Basepair` class so that bifurcations are considered and the application correctly finds the structure (even if it contains one or more bifurcations) with the maximum basepairings. You do not need to modify any code except the `computeStructureWithMaximumBasepairs` method in the `RNA_Basepair` class.

For the provided tRNA sequence, how many basepairs does the structure (*including* bifurcations) with maximum basepairings contain?

Does the tRNA structure with maximum basepairings correspond exactly to the putative tRNA structure illustrated here (<http://cs.wellesley.edu/~cs313/exercises/Exercise9/tRNA.jpg>)?

For the provided MiR-96 microRNA precursor sequence, how many basepairs does the structure (*including* bifurcations) with maximum basepairings contain?

Does the mir-96 structure with maximum basepairings correspond exactly to the putative mir-96 structure illustrated here (<http://cs.wellesley.edu/~cs313/exercises/Exercise9/mir-96.jpg>)?

Task 3: RNA Structure with Minimum Energy

Given an RNA sequence, rather than compute the structure with the maximum number of basepairings we could compute the structure with the minimum energy (lower energies are less costly and more favorable). Below are two recurrences that can be used for estimating the structure of minimum energy for an RNA sequence.

$$V(i, j) = \begin{cases} e_{hairpin}(i, j) & // \text{ hairpin loop} \\ e_{stacking}(i, j, i+1, j-1) + V(i+1, j-1) & // \text{ stacking region} \\ \min_{i+1 < i' < j' < j-1} (e_{interior}(i, j, i', j') + V(i', j')) & // \text{ interior loop} \\ \min_{i+1 < j' < j-1} (e_{bulge}(i, j, i+1, j') + V(i+1, j')) & // \text{ bulge loop (3')} \\ \min_{i+1 < i' < j-1} (e_{bulge}(i, j, i', j-1) + V(i', j-1)) & // \text{ bulge loop (5')} \\ \min_{i+1 < i' < j-2} (W(i+1, i') + W(i'+1, j-1)) & // \text{ bifurcation} \end{cases}$$

$$W(i, j) = \begin{cases} W(i+1, j) & // i \text{ doesn't pair} \\ W(i, j-1) & // j \text{ doesn't pair} \\ V(i, j) & // i \text{ and } j \text{ pair with each other} \\ \min_{i < i' < j-1} (W(i, i') + W(i'+1, j)) & // i \text{ and } j \text{ pair but not with each other} \end{cases}$$

In the recurrences above, $e_{hairpin}$, $e_{stacking}$, $e_{interior}$, and e_{bulge} refer to the energies of hairpin loops, stacking regions, interior loops, and bulge loops, respectively. The variables i and j represent indices of the RNA sequence, $V(i, j)$ represents the minimum energy of the subsequence between indices i and j , inclusive, such that the nucleotides at indices i and j basepair with each other, and $W(i, j)$ represents the minimum energy of the subsequence between indices i and j , inclusive, such that the nucleotides at indices i and j may or may not basepair.

For a given RNA sequence, the structure with minimum energy can be estimated via dynamic programming by filling in the two tables V and W , corresponding to the two recurrences above.

For an RNA sequence of length n , what is the running time in big-Oh notation of such a dynamic programming algorithm?

For an RNA sequence of length n , what is the running time in big-Oh notation of such a dynamic programming algorithm if only structures with loops less than 30 nucleotides in length are considered?

Task 4: Computational Identification of RNA Structure with Minimum Energy

Download the `RNA_Structure` folder from the `download` folder on the CS server. In this task, you will be working with the `RNA_Energy` application. Compile and execute the `RNA_Energy` application. The application requires one command line argument corresponding to a FASTA formatted file containing an RNA sequence. You can find some example files containing RNA sequences in the `sequences` sub-folder that you downloaded.

Given an RNA sequence, the `RNA_Energy` application computes the secondary structure that has the minimum energy. In the program's output, note the parenthesis-dot notation above the RNA sequence. Parenthesis-dot notation is used to indicate a secondary structure. A dot (i.e., a period) indicates the nucleotide does not pair in the structure. An open parenthesis indicates the nucleotide does pair with the corresponding closed parenthesis nucleotide in the structure.

Unfortunately, the `RNA_Energy` application is incomplete. When searching for the structure with the minimum energy, it does not consider structures with interior loops. Thus, if the structure with the minimum energy contains an interior loop, the `RNA_Energy` application will not identify the structure but rather report a suboptimal structure.

For the provided MiR-96 microRNA precursor sequence, what is the energy of the structure (*excluding* interior loops) with minimum energy?

As context, the `mir-96` microRNA is found in many animals including humans. It is a noncoding RNA gene that acts by regulating the expression of other genes. It is expressed specifically in the inner ear and adult eye.

Your job is to modify the `computeStructureWithMinimumEnergy` method in the `RNA_Energy` class so that interior loops are considered and the application correctly finds the structure (even if it contains one or more interior loops) with the minimum energy. You do not need to modify any code except the `computeStructureWithMinimumEnergy` method in the `RNA_Energy` class.

For the provided MiR-96 microRNA precursor sequence, what is the energy of the structure (*including* interior loops) with minimum energy?

Does the `mir-96` structure with minimum energy correspond exactly to the putative `mir-96` structure illustrated here (<http://cs.wellesley.edu/~cs313/exercises/Exercise9/mir-96.jpg>)?