

Atividade avaliativa 11

Atualizado automaticamente a cada 5 minutos

	<p>Introdução à Ciência da Computação II SSC0503</p> <p>Professor: Adenilso da Silva Simão PAE: Rodrigo Henrique Ramos Monitor: Daniel Martins</p>
--	--

Atividade avaliativa 11
Algoritmo Guloso
Data de entrega: 10/01/2022

Instruções:

Os trabalhos devem ser entregues no run.codes, na disciplina com código 6MDC. A atividade será realizada de **forma individual ou em grupos de até 4 alunos**. Todos os integrantes devem submeter o mesmo código. O grupo deve se "reunir" ao menos uma vez pelo google meet para discutir/resolver o problema e deve gravar a sessão. Além do código, **deve ser postado por apenas um integrante do grupo** o link para o vídeo da sessão gravada (no máximo 5 minutos) no fórum específico no e-disciplinas, **juntamente com a indicação dos integrantes do grupo**. O vídeo deve estar aberto para visualização de todos os



Atividade avaliativa 11

Atualizado automaticamente a cada 5 minutos

De uma maneira bastante simplificada, pode-se dizer que o sequenciamento do DNA é realizado a partir da leitura de vários fragmentos (chamadas de *reads*), que são “montadas” posteriormente.

Nessa atividade vamos simular a montagem do DNA a partir das *reads* sequenciadas. Para isso, faça um programa em C que leia um número inteiro N que indica a quantidade de *reads*, seguido da leitura de cada uma das *reads*. Após isso, utilize algoritmo guloso para montar essas *reads*, gerando como resultado apenas uma sequência. A estratégia do algoritmo guloso deve considerar a melhor sobreposição (*overlap*) entre as *reads* em cada etapa.

Exemplo:

Considere o conjunto seguinte de *reads*:

- `READS = [atccat, ctgata, ccatg]`

Primeira iteração das permutações das *reads* e possíveis montagens:

- `atccat + ctgata -> atccatctgata -> overlap 0`
- `atccat + ccatg -> atccatg -> overlap 4`
- `ctgata + atccat -> ctgatccat -> overlap 3`
- `ctgata + ccatg -> ctgatccatg -> overlap 1`
- `ccatg + atccat -> ccatgatccat -> overlap 0`
- `ccatg + ctgata -> ccatgctgata -> overlap 0`

Portanto, o maior overlap é de tamanho 4. Remove-se as sequências envolvidas nesta montagem e insere a nova sequência montada na primeira posição da lista de *reads*, tendo como resultado:

- `READS = [atccatg, ctgata]`

Repete-se o procedimento com a nova lista gerada



Atividade avaliativa 11

Atualizado automaticamente a cada 5 minutos

primeira encontrada.

3. Caso não aconteça sobreposição ($\text{overlap} = 0$), deve-se realizar a montagem das reads mesmo assim, concatenando as reads sem sobreposição.
4. Sobreposições completas no interior das *reads* também devem ser consideradas. Por exemplo, a combinação das reads “tcg” e “actcgaac” tem *overlap* 3 e a sequência resultante é a própria sequência “actcgaac”, pois “tcg” é substring de “actcgaac”.

Exemplo de entrada e saída:

Entrada	Saída
3 atccat ctgatc ccatg	ctgatccatg