

INSTITUTO SUPERIOR TÉCNICO
Análise e Síntese de Algoritmos
2025/2026

1º Projecto

Data enunciado: 10 de Novembro de 2025
Data Limite de Entrega: 5 de Dezembro de 2025

Descrição do Problema

O João Caracol vai participar num jogo de realidade virtual, onde os concorrentes retiram aminoácidos de uma cadeia de n aminoácidos a_1, \dots, a_n de uma proteína. Cada vez que um aminoácido a_i é retirado, uma certa quantidade de energia é libertada, que depende:

- da energia potencial P_i do aminoácido a_i ;
- da fixação do aminoácido aos dois aminoácidos mais próximos, dada pela energia potencial destes P_{i-1} e P_{i+1} ;
- e da afinidade $Af(C(a), C(b))$ entre a classe bioquímica Polar (P), Não-Polar (N), Ácido (A), ou Base (B), do aminoácido a com o aminoácido b .

e é dada pela seguinte expressão:

$$E_{libertada} = P_{i-1} \times Af(C(i-1), C(i)) \times P_i + P_i \times Af(C(i), C(i+1)) \times P_{i+1}$$

e onde a afinidade (≥ 0) entre duas classes é dada pela seguinte tabela (não simétrica!):

Af	P	N	A	B
P	1	3	1	3
N	5	1	0	1
A	0	1	0	4
B	1	3	2	3

As extremidades da cadeia, posição 0 e $n+1$, são consideradas pontos de fixação com uma energia potencial de 1 e uma classe bioquímica especial Terminal (T), que é sempre neutra em qualquer interacção, i.e., $Af(T, c) = Af(c, T) = 1$, qualquer que seja a classe c .

Assim, o objectivo é encontrar a ordem pelo qual se devem retirar todos os a_1, \dots, a_n , tal que o somatório da energia total libertada seja máximo, e ajudar o João Caracol a ganhar o jogo.

Input

O ficheiro de entrada contém toda a informação sobre a sequência de aminoácidos. Assim, o ficheiro de entrada é definido da seguinte forma:

- Uma linha contendo um inteiro $n \geq 1$ correspondendo ao número de aminoácidos na sequência;
- Uma linha com uma sequência de n inteiros, separados por espaço, correspondendo ao potencial (> 0) de estabilidade de cada aminoácido;
- Uma linha com uma sequência de n caracteres (sem espaços), correspondendo à classe bioquímica de cada aminoácido (P, N, A, ou B).

Output

O programa deverá examinar a sequência de aminoácidos e determinar a ordem pela qual devem ser retirados que maximiza a energia total libertada.

O output deve ser o seguinte:

- A primeira linha deve conter o valor da energia total libertada;
- A segunda linha deve conter a sequência de aminoácidos, separados por espaço, que permitiu esse valor de energia total libertada. Se existir mais do que uma sequência ótima, deve ser apresentada a que for lexicograficamente menor.

Exemplo 1

Input

```
3
10 5 12
ABA
```

Output

```
359
1 2 3
```

Exemplo 2

Input

```
9
4 2 7 3 5 1 2 8 3
ANBPAPBNA
```

Output

```
607
2 1 4 6 5 7 8 9 3
```

Implementação

A implementação do projecto deverá ser feita preferencialmente usando a linguagem de programação C++. Submissões nas linguagens Java/Python também serão aceites, embora fortemente desaconselhadas. Alunos que o escolham fazer devem estar cientes de que submissões em Java/Python podem não passar todos os testes mesmo implementando o algoritmo correcto. Mais se observa que soluções recursivas podem esgotar o limite da pilha quando executadas sobre os testes de maior tamanho, pelo que se recomenda a implementação de algoritmos **iterativos**.

O tempo necessário para implementar este projecto é inferior a 15 horas.

Parâmetros de compilação:

```
C++: g++ -std=c++11 -O3 -Wall file.cpp -lm
C: gcc -O3 -ansi -Wall file.c -lm
Javac: javac File.java
Java: java -Xss32m -Xmx256m -classpath . File
Python: python3 file.py
Rust: rustc -C opt-level=3 --edition=2021 file.rs
```

Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Informação sobre as linguagens de programação possíveis está disponível no website do sistema Mooshak. A linguagem de programação é identificada pela extensão do ficheiro. Por exemplo, um projecto escrito em c deverá ter a extensão `.c`. Após a compilação, **o programa resultante deverá ler do standard input e escrever para o standard output.**

Informação sobre as opções e restrições de compilação podem ser obtidas através do botão help do sistema Mooshak. O comando de compilação não deverá produzir output, caso contrário será considerado um erro de compilação.

Relatório: deverá ser submetido através do sistema Fénix no formato PDF com não mais de 2 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma descrição da solução, a análise teórica e a avaliação experimental dos resultados. O relatório deverá incluir qualquer referência que tenha sido utilizada na realização do projecto. Relatórios que não sejam entregues em formato PDF terão nota 0. Atempadamente será divulgado um template do relatório.

Código fonte: deve ser submetido através do sistema Mooshak e o relatório (em formato PDF) deverá ser submetido através do Fénix. O código fonte será avaliado automaticamente pelo sistema Mooshak (<http://acp.tecnico.ulisboa.pt/~mooshak/>). Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas: tal inclui o código fonte e o relatório.

Avaliação

O projecto deverá ser realizado em grupos de um ou dois alunos e será avaliado em duas fases. Na primeira fase, durante a submissão, cada implementação será executada num conjunto de testes, os quais representam 85% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 15% da nota final.

Avaliação Automática

A primeira fase do projecto é avaliada automaticamente com um conjunto de testes, os quais são executados num computador com o sistema operativo **GNU/Linux**. É essencial que o código fonte compile sem erros e respeite os standards de entrada e saída indicados anteriormente. Os projectos que não respeitem os formatos especificados serão penalizados e poderão ter nota 0, caso falhem todos os testes. Os testes **não serão divulgados antes da submissão**. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projecto. Além de verificar a correcção do output produzido, o ambiente de avaliação **restringe a memória e o tempo de execução** disponíveis. A maior parte dos testes executa o comando `diff` da forma seguinte:

```
diff output result
```

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `output` contém o output esperado. Um programa passa num teste e recebe o valor

correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output). O sistema reporta um valor entre 0 e 170.

A nota obtida na classificação automática poderá sofrer eventuais cortes caso a análise do código demonstre recurso a soluções ajustadas a inputs concretos ou outputs aleatórios/constantes.

Detecção de Cópias

A avaliação dos projectos inclui um procedimento para detecção de cópias. A submissão de um projecto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a reprovação na unidade curricular, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em “Diário da República”.