UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
PROJETO E ANÁLISE DE ALGORITMOS

1º semestre de 2024
Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 2
Data de Entrega: 14 de Maio 2024. Trabalho Dupla

Introdução

João não gosta de sentir tédio. É por isso que sempre que fica entediado, ele inventa jogos. Em uma longa noite de inverno, ele inventou um jogo e decidiu jogá-lo.

Dada uma sequência a composta por n inteiros. O jogador pode fazer várias jogadas. Em cada jogada ele pode escolher um elemento da sequência (vamos denotá-lo a_k) e excluí-lo, sendo que os elementos a_{k+1} e a_{k-1} também devem ser excluídos da sequência. Essa jogadas traz a_k pontos para o jogador.

João é um perfeccionista, então decidiu conseguir o máximo de pontos possível. Para isso João pediu para você criar um programa que mostre a qual a pontuação máxima que é possível atingir em cada jogo.

Assim, seu objetivo nesse trabalho é implementar duas estratégias para solucionar esse problema:

- Uma estratégia por Programação Dinâmica;
- Uma estratégia alternativa que seja capaz de sempre resolver o problema;

Entrada e Saída

A primeira linha contém o inteiro N ($0 \leq N \leq 10^5$) que mostra quantos números existem na sequência de Joâo.

A segunda linha contém N inteiros $a_1, a_2, ..., a_n$.

Exemplo de entrada:

```
9
1 2 1 3 2 2 2 2 3
```

Exemplo de saída

```
10
```

Seu executável deve chamar tp2 e será chamado da seguinte forma:

```
./tp2 <estrategia> entrada.txt Onde:
```

- <estrategia> é D para a solução baseada em programação dinâmica e A para a solução alternetiva;

Seu programa deve criar um arquivo **saida.txt** com a resposta, na tela, o programa deve imprimir apenas os tempos de usuário e os tempos de sistema para comparação. Para avaliação do tempo, utilize as funções *getrusage* e *gettimeofday*.

Documentação

Deve ser clara e objetiva, descrevendo as soluções adotadas e justificando bem as escolhas realizadas. Devem possuir também uma análise de complexidade detalhada das soluções. Em termos de análise de resultados, avalie o desempenho e funcionamento de seus algoritmos para diversas configurações e avalie também o tempo de execução dos mesmos (compare-os). Lembre-se, o importante é você apresentar maturidade técnica em suas discussões.

Observações:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios do DCOMP.
- Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado utilizando os arquivos .c .h.
- O utilitário Make deve ser utilizado para compilar o programa;
- A saída deve ser impressa no arquivo pedido seguindo estritamente o formato da especicação caso contrário o resultado será considerado errado;
- O arquivo executável deve ser chamado de **tp2** e deve receber como parâmetro apenas o nome do arquivo de entrada de dados e a estratégia escolhida pelo usuario. Não serão aceitos outros nomes de executáveis além dos mencionados.
- Faça seu código de forma legível

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- documentação contendo:;
  - descrição das soluções e estruturas de dados utilizadas;
  - análise da complexidade das rotinas;
  - análise dos resultados obtidos.
  - a documentação não pode exceder 12 páginas.

Distribuição dos pontos:

- execução (E)

  execução correta: 80%

- estilo de programação
  código bem estruturado: 10%
  código legível: 10%
- documentação (D)
  comentários explicativos: 40%
  análise de complexidade: 30%
  análise de resultados: 30%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$