UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
ALGORITMOS E ESTRUTURAS DE DADOS III

1º semestre de 2016
Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 2
Data de Entrega: 20 de Abril 2016.

Strawberry Fields Forever

Você foi convidado por um amigo para resolver um problema de colheita de morangos. Considere uma plantação de morangos, representada por uma matriz de duas dimensões, onde cada campo dessa matriz possui um pezinho com um número específico de morangos. Se uma pezinho não possui morango, o campo é denotado por 0. Partindo de um dos pé de morango mais à esquerda (quadrantes mais a esquerda da matriz), e parando em um pé mais à direita (quadrantes mais a direita), considerando que você só pode mexer para cima, para baixo e para direita e que cada pé de morango só pode ser vizitado uma vez, você precisa fazer um algoritmo que determine o máximo de morangos que é possível coletar. Para tal tarefa, considere as seguintes condições:

- Pés de morango são muito sensíveis, assim você não pode percorrer pés sem morango;
- Mesmo estando nas bordas à esquerda ou à direita, você pode caminhar para baixo ou para cima;
- Quando você atingir alguma célula na borda do topo, você pode continuar caminhando para cima, sendo que você será teletransportado para a borda de baixo. O mesmo se aplica a borda inferior, sendo você teletransportado para borda superior. Entretanto isso tem um custo que será todos os morangos correntes.

Para embalar suas horas de programação, segue uma interessante sugestão:
https://www.youtube.com/watch?v=8UQK-UcRezE

Entrada e Saída

A entrada é composta, na primeira linha pelas dimensões do campo. As seguintes linhas representam o total de morangos de cada célula, ao longo de cada linha e cada coluna. A saída é o número máximo de morangos que é possivel coletar.

Exemplo 1 entrada:

```
4 4
0 4 5 1
2 0 2 4
3 3 0 3
4 2 1 2
```

Exemplo de Saída: 23

Exemplo 2 entrada:

```
4 4
0 4 5 1
2 0 2 4
3 3 0 0
4 2 1 2
```

Exemplo de Saída: 22

O arquivo executável deve ser chamado de tp2 e deve receber como parâmetro o arquivo de entrada e o arquivo de saída

```
./tp2 -i entrada.txt -o saida.txt
```

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 seu algoritmo para diversas configurações e avalie também o tempo de execução dos mesmos. 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 **tp1** e deve receber como parâmetro apenas o nome do arquivo de entrada de dados. 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):

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