

Instituto de Computação - Unicamp

MC102 - Algoritmos e Programação de Computadores

Laboratório 10 - O Lobo de Wall Street

Prazo de entrega: **21/04/2017 23:59:59**

Peso: **1**

Professor: Eduardo C. Xavier

Professor: Guido Araújo

Monitor: Arthur Pratti Dadalto

Monitor: Cristina Cavalcante

Monitor: Klairton de Lima Brito

Monitor: Luís Felipe Mattos

Monitor: Paulo Finardi

Monitor: Paulo Lucas Rodrigues Lacerda

Monitor: Pedro Alves

Monitor: Renan Vilas Novas

Descrição



A corretora Stratton Oakmont, é conhecida por usar diversos métodos fraudulentos para operar na bolsa de valores, e você é um dos únicos funcionários honestos da corretora. Para demonstrar que é possível ganhar dinheiro de forma honesta, baseado em técnicas estatísticas e computacionais, você cria um programa que faz a previsão dos valores das ações da bolsa com precisão segura para alguns dias a frente. Este sistema é capaz de prever o valor de cada ação de um conjunto de empresas em cada um dos dias de um período no futuro.

Para concluir o sistema, ficou faltando apenas criar um programa que determina as decisões de compra e venda, ou seja, para cada dia deve ser determinado se é para ser feita uma operação de compra ou venda e de qual empresa.

O programa está em fase experimental e por conservadorismo deve escolher os dias de compra, ou venda, e de qual empresa, respeitando as seguintes regras:

- Somente quatro empresas terão ações disponíveis para compra/venda;
- A compra e venda se dá por um bloco de ações inteiras, e a entrada do programa contém exatamente o valor do bloco de ações de cada empresa em cada dia;

- A carteira de aplicações deve possuir a cada dia ações de no máximo uma empresa, ou seja, para comprar ações de uma empresa deve-se vender as ações previamente compradas de outra empresa;
- Pode-se comprar e vender ações de cada empresa no máximo uma vez no período avaliado, ou seja, após vendido o bloco de ações de uma empresa, não se pode mais realizar a compra de uma ação dela no futuro;
- A compra de um bloco de ações é feita no fim do dia e a venda é feita no começo, de modo que pode-se vender as ações de uma empresa e comprar a de outra no mesmo dia;

O objetivo é determinar a compra/venda de cada bloco de ações visando maximizar o lucro, sendo o ganho com cada bloco de ações definido como o valor que se vendeu aquele bloco menos o valor que ele foi comprado.

Por exemplo, considere a entrada abaixo listando os valores das ações de quatro empresas (uma linha para cada empresa) durante três dias (uma coluna para cada dia).

211.0	120.0	130.0
21.00	22.00	23.00
310.0	320.0	33.00
41.00	42.00	43.00

A melhor solução é comprar o bloco de ações da empresa 3 no dia 1 (por 310.0) e vender no dia 2 (por 320.0), e comprar o bloco de ações da empresa 1 no dia 2 (por 120.0) e vender no dia 3 (por 130.0) obtendo um lucro de 20.

Objetivo

O seu objetivo é fazer um programa que avalia os valores diários dos blocos de ações de quatro empresas e decide qual é o melhor dia para comprar e vender as ações de cada empresa, atentando para as regras de que ele pode ter na carteira de aplicações um bloco por vez e as ações de cada empresa podem ser compradas no máximo uma vez. O número

máximo de dias na entrada será 15.

O seu programa deve determinar qual(is) bloco(s) de ações comprar/vender de tal forma a se maximizar o lucro.

Entrada

A entrada consiste em um valor inteiro d representando a quantidade de dias no período analisado, seguido de d valores reais para cada uma das quatro empresas avaliadas, representando os valores do bloco de ações da empresa em cada dia, começando no dia 1 até o dia d .

Por exemplo, se a entrada for igual a

```
3
211.0 120.0 130.0
21.00 22.00 23.00
310.0 320.0 33.00
41.00 42.00 43.00
```

significa que serão avaliados 3 dias, a ação da primeira empresa no primeiro dia vale 211, no segundo dia vale 120 e no terceiro dia vale 130; a ação da segunda empresa no primeiro dia vale 21, no segundo dia vale 22 e no terceiro dia vale 23; a ação da terceira empresa no primeiro dia vale 310, no segundo dia vale 320 e no terceiro dia vale 33; a ação da quarta empresa no primeiro dia vale 41, no segundo dia vale 42 e no terceiro dia vale 43.

Saída

Como saída o seu programa deverá imprimir para cada empresa de 1 até 4, nesta ordem, as informações:

acao N: compra DC, venda DV, lucro LC

onde N é o número da empresa (de 1 até 4), DC o dia da compra

realizada, dv o dia da venda, e lc o lucro com 2 casas decimais. **OBS:** Só deverá ser impresso informações de compra/venda para uma empresa se efetivamente foi feita uma operação com a empresa.

Por fim deve ser impresso o lucro total:

```
Lucro: LC
```

onde LC deve ser o lucro total obtido com 2 casas decimais.

Se não for realizada nenhuma operação de compra/venda com nenhuma empresa o programa deverá imprimir apenas:

```
Lucro: 0.00
```

Uma saída correta para o exemplo anterior é

```
acao 1: compra 2, venda 3, lucro 10.00
acao 3: compra 1, venda 2, lucro 10.00
Lucro: 20.00
```

Você poderá usar o comando abaixo para imprimir a relação de compra e venda:

```
printf("acao %d: compra %d, venda %d, lucro %.2f\n", empresa, dia_cor
```

E o comando abaixo para imprimir o lucro

```
printf("Lucro: %.2f\n", lucro_total);
```

Dica de Implementação

Para cada empresa i seja c_i (v_i) uma variável que indica o dia de compra (venda) do bloco de ações da empresa. Se $c_i = 0$ assume-se que

não haverá compra para esta empresa e se c_i estiver em $[1, d]$ haverá compra com venda em $v_i > c_i$. Você pode gerar todas as possibilidades de dias de compra/venda para as 4 empresas usando laços encaixados, mas estes dias só serão válidos se não houver interseção entre os dias de compra/venda de empresas diferentes. Então para os intervalos gerados (c_i, v_i) em que $c_i \neq 0$, basta testar se são disjuntos, e ao longo da geração dos intervalos armazenar aqueles que dão o maior lucro.

Exemplos

Teste 01

Entrada

```
3
211.0 120.0 130.0
21.0 22.0 23.0
310.0 320.0 33.0
41.0 42.0 43.0
```

Saída

```
acao 1: compra 2, venda 3, lucro 10.00
acao 3: compra 1, venda 2, lucro 10.00
Lucro: 20.00
```

Teste 02

Entrada

```
3
120.0 110.0 100.0
140.0 130.0 100.0
150.0 120.0 100.0
200.0 100.0 50.0
```

Saída

Lucro: 0.00

Teste 03

Entrada

```
6
100.2998225252948 65.99007576793781 79.423150054003:
103.90895304277787 79.62634870562545 76.52331260424:
88.50765515692655 88.22684843983303 69.213555954912:
106.69097138255891 135.4956300660518 150.8966532089:
```

Saída

```
acao 1: compra 3, venda 5, lucro 98.01
acao 3: compra 5, venda 6, lucro 19.43
acao 4: compra 1, venda 3, lucro 44.21
Lucro: 161.65
```

Para mais exemplos, consulte os [testes abertos no Susy](#).

Observações

- O número máximo de submissões é **10**;
- O seu programa deve estar completamente contido em um único arquivo denominado `lab10.c` ;
- Para a realização dos testes do SuSy, a compilação dos programas desenvolvidos considerar o comando: `__ gcc -std=c99 -pedantic -Wall -o lab10 lab10.c ;`
- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA;
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.

Critérios importantes

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará em nota zero nesta tarefa de laboratório.

- O único header aceito para essa tarefa é o `stdio.h` .