



Universidade de Brasília

Departamento de Ciência da Computação



Semana 10

Revisão 1

CIC0004

Algoritmos e Programação de Computadores

Prof. Pedro Garcia Freitas

<https://pedrogarcia.gitlab.io/>

pedro.garcia@unb.br

Brasília



Este conjunto de slides não deve ser utilizado ou republicado sem a expressa permissão do autor.

This set of slides should not be used or republished without the author's express permission.



1. Você foi contratado para ser desenvolvedor de software de um banco. Em sua primeira atribuição, ficou responsável por escrever um programa que calcule e entregue as notas de um saque. Seu programa deve receber como entrada um valor a ser sacado e retornar a quantidade de notas de R\$100, R\$50, R\$20, R\$10, R\$2 e R\$1 necessárias para efetuar o saque. Considere que seu programa está em processo de homologação e, por isso, não irá disponibilizar as notas fisicamente, mas exibirá na tela do time de testes a quantidade de notas que ele irá retornar. Seguem os exemplos abaixo:

Exemplo 01:

```
Valor do saque: R$ 120  
Entregue 1 nota de R$ 100,00  
Entregue 1 nota de R$ 20,00
```

Exemplo 02

```
Valor do saque: R$ 4  
Entregue 2 notas de R$ 2,00
```

Exemplo 03:

```
Valor do saque R$ 666  
Entregue 6 notas de R$ 100,00  
Entregue 1 notas de R$ 50,00  
Entegue 1 nota de R$ 10,00  
Entregue 3 nota de R$ 2,00
```



2. Interpretador louco: Você ficou encarregado de escrever um interpretador que realiza operações binárias entre dois números inteiros de 8 bits. Seu interpretador possui uma *instruction set architecture* reduzida, contendo apenas as instruções binárias XOR, OR, e AND. Contudo, antes de realizar essas operações o seu interpretador inverte um bit aleatoriamente nos números de entrada. Escreva um programa que recebe dois números inteiros, converta-os para representação binária, inverta um bit aleatório desses operandos e realize as operações XOR, OR, e AND. Por fim, exiba os números lidos, suas representações binárias, os bits invertidos, os novos números gerados a partir desses bits invertidos e o valor das operações.

Exemplo	Input	Output
1	10 20	inputs: 10 20 binary equivalents: 00001010 00010100 inverted equivalents: 10001010 00110100 binary diffs: 10000000 00100000 XOR(10001010,00110100)=10111110 OR(10001010,00110100)=10111110 AND(10001010,00110100)=00000000
2	90 77	inputs: 90 77 binary equivalents: 01011010 01001101 inverted equivalents: 01010010 01001111 binary diffs: 00001000 00000010 XOR(01010010,01001111)=00011101 OR(01010010,01001111)=01011111 AND(01010010,01001111)=01000010
3	255 0	inputs: 255 0 binary equivalents: 11111111 00000000 inverted equivalents: 11111110 10000000 binary diffs: 00000001 10000000 XOR(11111110,10000000)=01111110 OR(11111110,10000000)=11111110 AND(11111110,10000000)=10000000

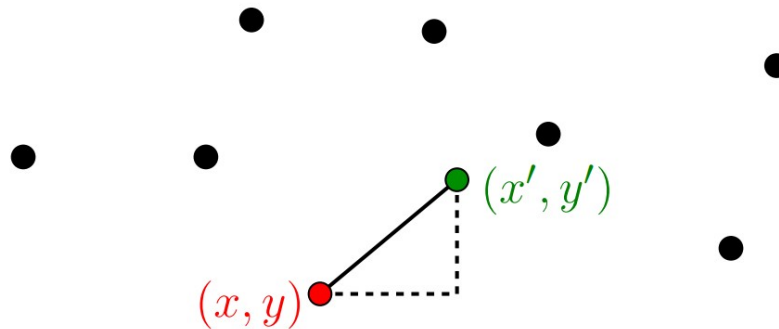


3. UnBCab. Você foi contratado para criar um aplicativo de transporte que permite a busca por motoristas baseada na localização, oferecendo um serviço semelhante ao tradicional táxi, mas onde o custo da passagem é dividido entre os passageiros. No seu primeiro dia de trabalho, você ficou encarregado de escrever o programa responsável por calcular o preço da passagem. No modelo de negócio da empresa, considera-se o consumo de combustível por km, o número de passageiros (e a tarifa por passageiro) e a distância em km percorrida. Nesse modelo, considere que o preço do combustível como sendo R\$ 5.00/L, o consumo é de 2 L/km, e cada passageiro paga uma taxa mínima de R\$ 4.00, independente da distância. Além disso, deve-se considerar a taxa da operação financeira, que depende do banco do pagador. Essa taxa é um número real entre 0 (0%) e 1 (100%). Sendo assim, tomando como entrada o número de passageiros, a distância do trajeto e a taxa do banco, determine o preço da viagem.

Exemplo	Input	Output
1	Quantos usuários (1 a 4): 1 Distância a ser percorrida (em KMs): 30 Digite a taxa do banco (entre 0 e 1): 0.1	O preço final da viagem é: 334.40
2	Quantos usuários (1 a 4): 3 Distância a ser percorrida (em KMs): 12 Digite a taxa do banco (entre 0 e 1): 0.1	O preço final da viagem é: 57.20
3	Quantos usuários (1 a 4): 3 Distância a ser percorrida (em KMs): 12 Digite a taxa do banco (entre 0 e 1): 0.25	O preço final da viagem é: 65.00
4	Quantos usuários (1 a 4): 1 Distância a ser percorrida (em KMs): 12 Digite a taxa do banco (entre 0 e 1): 0.25	O preço final da viagem é: 155.00
5	Quantos usuários (1 a 4): 2 Distância a ser percorrida (em KMs): 12 Digite a taxa do banco (entre 0 e 1): 0.25	O preço final da viagem é: 85.00

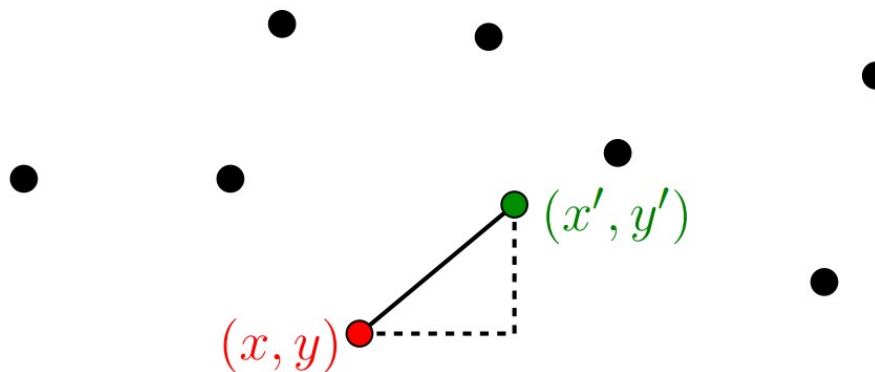


4. Em várias aplicações envolvendo geometria, principalmente no desenvolvimento de jogos em Python, é comum nos depararmos com a necessidade de calcular a distância entre dois pontos A e B. Também quando estamos trabalhando com computação gráfica e jogos, muitas vezes é necessário calcular a velocidade e/ou o tempo de deslocamento para um objeto sair de um ponto A para um ponto B. Sendo assim, considere o problema: Dados n pontos no plano, determinar dois deles que estão à distância mínima.





4. Problema: Dados n pontos no plano, determinar **dois deles** que estão à distância mínima.



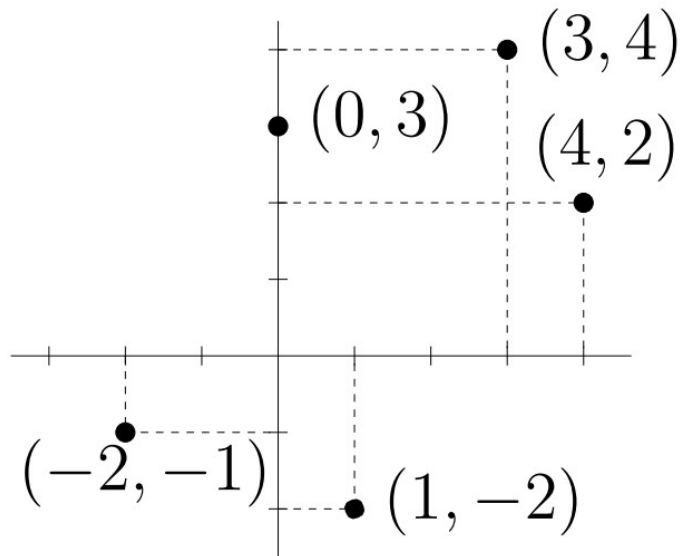
Lembre-se que, para dois pontos (x, y) e (x', y') no plano,

$$\text{DIST}(x, y, x', y') = \sqrt{(x - x')^2 + (y - y')^2}.$$



4. Problema: Dados n pontos no plano, determinar **dois deles** que estão à distância mínima.

Entrada: coleção de n pontos representada pelos $X[1..n]$ e $Y[1..n]$.

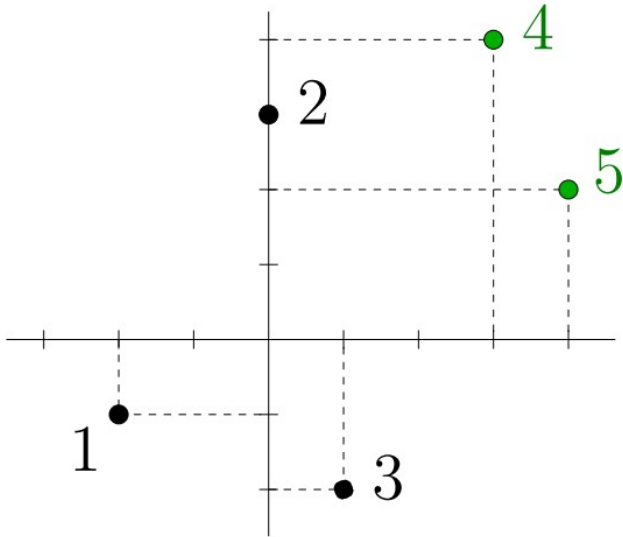


X	-2	0	1	3	4
Y	-1	3	-2	4	2
	1	2	3	4	5



4. Problema: Dados n pontos no plano, determinar **dois deles** que estão à distância mínima.

Entrada: coleção de n pontos representada pelos $X[1..n]$ e $Y[1..n]$.



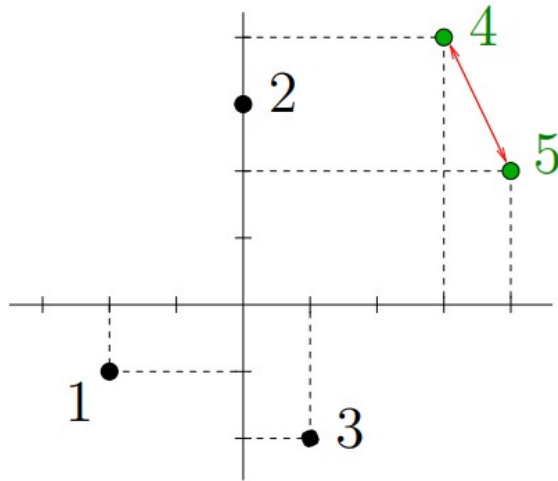
X	-2	0	1	3	4
Y	-1	3	-2	4	2
	1	2	3	4	5

Saída: índices i e j indicando dois pontos à distância mínima.



4. Problema: Dados n pontos no plano, determinar **dois deles** que estão à distância mínima.

Entrada: coleção de n pontos representada pelos $X[1..n]$ e $Y[1..n]$.




X	-2	0	1	3	4
Y	-1	3	-2	4	2
	1	2	3	4	5

Saída: **menor distância** entre dois pontos da coleção.

Exemplo	Input	Output
1	Digite a quantidade de pontos: 5 Coordenadas do ponto 1: -2 -1 Coordenadas do ponto 2: 0 3 Coordenadas do ponto 3: 1 -2 Coordenadas do ponto 4: 3 4 Coordenadas do ponto 5: 4 2	Os pontos mais próximos são: Ponto 4 (3,4) e Ponto 5 (4,2) Distancia=2.24
2	Digite a quantidade de pontos: 3 Coordenadas do ponto 1: 0 1 Coordenadas do ponto 2: -1 0 Coordenadas do ponto 3: 2 3	Os pontos mais próximos são: Ponto 1 (0,1) e Ponto 2 (-1,0) Distancia=1.41
3	Digite a quantidade de pontos: 2 Coordenadas do ponto 1: -1 0 Coordenadas do ponto 2: 0 1	Os pontos mais próximos são: Ponto 1 (-1,0) e Ponto 2 (0,1) Distancia=1.41
4	Digite a quantidade de pontos: 8 Coordenadas do ponto 1: -1 -1 Coordenadas do ponto 2: -2 -2 Coordenadas do ponto 3: 0 0 Coordenadas do ponto 4: 1 1 Coordenadas do ponto 5: 2 2 Coordenadas do ponto 6: 3 4 Coordenadas do ponto 7: -3 2 Coordenadas do ponto 8: -1 1	Os pontos mais próximos são: Ponto 1 (-1,-1) e Ponto 2 (-2,-2) Distancia=1.41



5. Você sabia que, dependendo do país, existem diferentes modos de escrever as datas? Não existe um modo universal em todas as nações para escrever datar os dias e anos, porém métodos mais utilizados de acordo com países. Por exemplo, aqui no Brasil nós escrevemos dia/mês/ano. Já nos Estados Unidos, o correto é escrever mês/dia/ano. Por outro lado, no Japão escrevemos ano/mês/dia. Enquanto no Canadá todos esses modos estão certos . 

O método aplicado é importante para bancos, para empresas, para calendários escolares e para nós organizarmos as nossas férias, porém não existem diferenças substanciais entre um modo e outro. Como cada país adota o sistema que julga ser mais interessante, é preciso chegar a um consenso global para evitar problemas relacionados às datas. Foi assim que chegamos ao **ISO 8601**, emitido pela Organização Internacional de Padronização.



5. A principal característica do formato de data e hora da norma ISO 8601 é que a informação da data e da hora seja ordenada a partir do valor mais significativo – em outras palavras, do maior ao menor (**ano/mês/dia**). Hoje, existem alguns países que adotaram o ISO 8601 como data oficial para facilitar essa padronização, como é o caso do Japão e Coreia do Sul.

Faça um programa que detecte qual o formato da data e a converta para o formato ISO 8601. Isto é, se ela tiver no formato US (mês/dia/ano), converta-a para ano/mês/dia. Se a data de entrada estiver no formato BR (dia/mês/ano), converta-a para ano/mês/dia. Se ela já tiver no formato ISO, apenas imprima na tela. Mas se não for possível determinar se a data está no formato BR ou US, imprima a mensagem de erro “Formato de entrada indeterminado”.

Exemplo	Input	Output
1	Digite uma data separada por '/': 2020/09/12	2020/09/12
2	Digite uma data separada por '/': 12/12/2023	Formato de entrada indeterminado!
3	Digite uma data separada por '/': 20/02/1964	1964/02/20
4	Digite uma data separada por '/': 02/15/2024	2024/02/15v



A word cloud centered around the word "THANK YOU". Other prominent words include "GRACIAS", "ARIGATO", "SHUKURIA", "JUSPAXAR", "DANKSCHEEN", "TASHAKKURATU", "YAQHANYELAY", "SUKSAMA", "EKHMET", "TINGKI", "BİYAN", "SHUKRIA", "GOZAIMASHITA", "BECHARISTO", "KOMAPSUMNDA", "MAAKE", "GRAZIE", "MEHRBANI", "PALMES", "BOLZİN", and "MERCİ". The words are arranged in various orientations and sizes, creating a dense, colorful collage.



Dúvidas?

