

Fundamentos da Programação

lista extra 01

1. Quem é igual a quem?

Imagine dentre as 5 variáveis abaixo

U	V	X	Y	Z
---	---	---	---	---

- duas delas armazenam o mesmo valor
- e as outras três armazenam um valor diferente (também igual para todas elas)

Faça um programa que descobre quem tem o mesmo valor que quem.

A ideia é primeiro resolver o problema, e depois reduzir ao máximo o número de comparações realizadas pelo programa.

- Faça uma versão do programa que não utiliza variáveis auxiliares.
- E depois veja se você consegue encontrar uma solução melhor utilizando variáveis auxiliares.

2. Quem tem o 1?

Imagine que as variáveis

U	V	X	Y	Z
---	---	---	---	---

armazenam todas o número 0, exceto uma delas que tem o número 1.

O problema consiste em descobrir onde está o número 1.

E a ideia, mais uma vez, é realizar o menor número de comparações possível.

A solução trivial resolve o problema com 5 comparações (no pior caso).

Mas é possível reduzir esse número para 3.

(Será que é possível resolver o problema com menos do que 3 comparações, no pior caso?)

3. Onde está o 1 (na lista)?

Agora considere a versão do problema anterior com uma lista de tamanho N

0	...	N-1
---	-----	-----

L

Quer dizer, todas as posições da lista contém o número 0, exceto por uma delas que contém o 1.

A tarefa consiste em descobrir a posição em que está o 1.

Quantas comparações são necessárias para resolver esse problema (no pior caso)?

4. Quem aparece mais vezes?

Considere uma lista de tamanho N que armazena números inteiros positivos

Por exemplo

L	3	5	2	5	1	3	3	4	2	4
---	---	---	---	---	---	---	---	---	---	---

Note que o mesmo número pode aparecer mais de uma vez na lista.

E a tarefa consiste em descobrir qual é o número que aparece mais vezes.

Faça um programa que resolve esse problema.

5. Pares primeiro, e ímpares depois

Faça um programa que reorganiza os números armazenados em uma lista, de modo que os pares apareçam primeiro e os ímpares apareçam depois.

Abaixo nós temos um pequeno exemplo dessa reorganização

3	2	6	9	4	8
---	---	---	---	---	---

 \Rightarrow

2	6	4	8	3	9
---	---	---	---	---	---

Nota: A ordem em que os números pares e ímpares aparecem no resultado não importa.

A única coisa importante é que os pares venham antes dos ímpares

— (*e que você não perca nenhum número ...*)

6. Soma 20

Imagine que você quer saber se a lista

	0	...	N-1
L			

contém dois números cuja soma é exatamente igual a 20.

Faça um programa que descobre se isso é o caso ou não.

7. Complicando um pouquinho as coisas

Agora imagine que você quer saber se a lista

	0	...	N-1
L			

contém 3 números tais que a soma de dois deles é exatamente igual ao terceiro.

Faça um programa que descobre se isso é o caso ou não.

8. Conversão de base

Imagine que B é uma lista de caracteres que contém apenas os símbolos 0 e 1.

Por exemplo,

B

1	0	0	1	0	1	1
---	---	---	---	---	---	---

Então, nós podemos interpretar o conteúdo de B como um número na base 2.

No exemplo acima, nós teríamos o seguinte

$$B = (1001011) = 75$$

- a) Faça um programa que converte o número binário armazenado em B para a base 10
- b) Faça um programa que converte o número inteiro armazenado na variável X para a base 2, e armazena o resultado em B.

9. Operando na base 2

Imagine que as variáveis B e C armazenam números binários

B

1	0	0	1	1
---	---	---	---	---

C

1	0	1	0
---	---	---	---

- a) Faça um programa que calcula a soma dos números binários armazenados em B e C, e armazena o resultado na variável D.
- b) Faça um programa que calcula o produto dos números binários armazenados em B e C, e armazena o resultado na variável D.

10. Operando com inteiros muito, muito longos

Nós já vimos que existe um limite para o tamanho dos números que podem ser armazenados em uma variável do tipo `int`.

E nós já vimos que as variáveis do tipo `long` podem armazenar muito maiores, mas elas também tem um limite.

Para ir além desse limite, nós podemos representar os nossos números como sequências de dígitos — (como foi feito no exercício anterior).

Por exemplo, abaixo nós temos um número bem grande armazenado na variável A

A

7	0	9	2	5	2	8	9	9	3	1	4	2	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---

- a) Faça um programa que soma dois inteiros muito, muito longos armazenados nas variáveis A e B, e armazena o resultado em C.
- b) Faça um programa que calcula o produto de dois inteiros muito, muito longos armazenados nas variáveis A e B, e armazena o resultado em C.

- c) Faça um programa que realiza a divisão de dois inteiros muito, muito longos armazenados nas variáveis A e B, e armazena o resultado em C.

11. Colocando as coisas em ordem

Imagine que a variável T armazena três nomes em uma ordem qualquer.

Por exemplo

T

D	e	d	é		Z	a	c	a	r	i	a	s		M	u	s	s	u	m
---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---

Faça um programa que coloca os nomes armazenados em T em ordem alfabética.

No exemplo acima, o seu programa deve produzir o seguinte resultado

T

D	e	d	é		M	u	s	s	u	m		Z	a	c	a	r	i	a	s
---	---	---	---	--	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---

12. Embaralhando as coisas (e desembaralhando depois)

Imagine que a variável T contém um texto com n palavras.

Uma ideia para embaralhar as palavras desse texto consiste em utilizar uma lista que contém uma permutação dos números 1 a n .

Por exemplo, para $n = 6$, nós teríamos algo como

P

5	2	4	1	6	3
---	---	---	---	---	---

E daí, a gente coloca a i -ésima palavra de T na posição indicada por $P[i]$ em uma outra variável R.

Por exemplo, ao embaralhar o seguinte texto

T

A		D	o	r	a		a	d	o	r	a		d	o	c	e		d	e		a	m	o	r	a
---	--	---	---	---	---	--	---	---	---	---	---	--	---	---	---	---	--	---	---	--	---	---	---	---	---

utilizando a permutação acima, nós obtemos

R

d	e		D	o	r	a		d	o	c	e		A		a	m	o	r	a		a	d	o	r	a
---	---	--	---	---	---	---	--	---	---	---	---	--	---	--	---	---	---	---	---	--	---	---	---	---	---

- Faça um programa que embaralha as n palavras do texto armazenado em T utilizando a permutação dos números 1 a n armazenada em P.
- E depois, faça um programa que desembaralha as palavras do texto armazenado em R, utilizando (apenas) a permutação P.