#### **Unidade I:**

#### Conceitos Básicos - Exercícios Iniciais

Prof. Max do Val Machado



Instituto de Ciências Exatas e Informática Curso de Ciência da Computação

# Potência de Dois

Para a aula de análise de complexidade, resolva as equações abaixo:

a) 
$$2^0 =$$

d) 
$$2^3 =$$

g) 
$$2^6 =$$

j) 
$$2^9 =$$

b) 
$$2^1 =$$

e) 
$$2^4 =$$

h) 
$$2^7 =$$

k) 
$$2^{10} =$$

c) 
$$2^2 =$$

f) 
$$2^5 =$$

i) 
$$2^8 =$$

$$1) 2^{11} =$$

# Logaritmo na Base Dois

Para a aula de análise de complexidade, resolva as equações abaixo:

a) 
$$lg(2048) = d$$
)  $lg(256) = g$ )  $lg(32) = g$ )  $lg(4) = g$ 

b) 
$$lg(1024) = e) lg(128) = h) lg(16) = k) lg(2) =$$

c) 
$$lg(512) = f$$
)  $lg(64) = i$ )  $lg(8) = i$ )  $lg(1) = i$ 

# Gráfico de Funções

Plote um gráfico com todas as funções abaixo:

a) 
$$f(n) = n$$

b) 
$$f(n) = n^2$$

c) 
$$f(n) = n^3$$

d) 
$$f(n) = sqrt(n)$$

e) 
$$f(n) = lg(n)$$

f) 
$$f(n) = 3n^2 + 5n - 3$$

g) 
$$f(n) = -3n^2 + 5n - 3$$

h) 
$$f(n) = |-n^2|$$

i) 
$$f(n) = 5n^4 + 2n^2$$

$$j) \quad f(n) = n * lg (n)$$

# Exercício

Faça um método que receba um array de inteiros e um número inteiro x e retorne um valor booleano informando se o elemento x está contido no array

 Repita o exercício acima considerando que os elementos do array estão ordenados de forma crescente. Uma sugestão seria começar a pesquisa pelo meio do array

## Exercício

 Faça um método que receba um array de inteiros e mostre na tela o maior e o menor elementos do array.

 Repita o exercício acima fazendo menos comparações com os elementos do array

#### O que o código abaixo faz?

```
public static boolean doidao (char n){
    boolean resp= false;
    int v = (int) c;
    if (v == 65 || v == 69 || v == 73 || v == 79 || v == 85 || v == 97 || v == 101 || v == 105 ||
        v == 111 || v == 117){
        resp = true;
    }
    return resp;
}
```

O que o código abaixo faz?

```
public static boolean doidao (char n){
    boolean resp= false;
    int v = (int) c;
    if (v == 65 || v == 69 || v == 73 || v == 79 || v == 85 || v == 97 || v == 101 || v ==105 ||
       V == 111 || V == 117){
        resp = true;
    return resp;
                                  Fazendo o isVogal, fica mais fácil ...
public static char toUpper(char c){
    return (c >= 'a' && c <= 'z') ? ((char) (c - 32)) : c;
public static boolean isVogal (char c){
    c = toUpper(c);
    return (c =='A' || c =='E' || c =='I' || c =='O' || c =='U');
```

#### Outras opções

```
public static boolean isLetra (char c){
    return (c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'Z');
}

public static boolean isConsoante (char c){
    return (isLetra(c) == true && isVogal(c) == false);
}</pre>
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
   if (n!=s.length()){
     if (s.charAt(n)<'0' || s.charAt(n)>'9'){
       if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
         s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
         s.charAt(n)=='o' || s.charAt(n)=='u'){}
         resp= false;
       } else{
         n++;
         resp=isConsoante(s, n);
     } else {
       resp=false;
   return resp;
```

```
public static boolean isConsoante(String s, int n){
   boolean resp= true;
   if (n!=s.length()){
     if (s.charAt(n)<'0' || s.charAt(n)>'9'){
       if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
         s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
         s.charAt(n)=='o' || s.charAt(n)=='u'){}
         resp= false;
       } else{
         n++;
         resp=isConsoante(s, n);
     } else {
                                   1º passo: Indentação
       resp=false;
   return resp;
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n != s.length()){
         if (s.charAt(n)<'0' || s.charAt(n)>'9'){
              if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
                s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
                s.charAt(n)=='o' || s.charAt(n)=='u'){}
                  resp= false;
              } else {
                   n++;
                  resp=isConsoante(s, n);
         } else {
              resp=false;
    return resp;
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n != s.length()){
         if (s.charAt(n)<'0' || s.charAt(n)>'9'){
              if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
                s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
                s.charAt(n)=='o' || s.charAt(n)=='u'){}
                  resp= false;
              } else {
                  n++;
                  resp=isConsoante(s, n);
         } else {
              resp=false;
    return resp;
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n != s.length()){
         if (s.charAt(n)<'0' || s.charAt(n)>'9'){
             if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
               s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
               s.charAt(n)=='o' || s.charAt(n)=='u'){}
                  resp= false;
             } else {
                  n++;
                  resp=isConsoante(s, n);
         } else {
                                   2º passo: Simplificação
             resp=false;
    return resp;
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n != s.length()){
         if (s.char\Delta t(n) < 0) \parallel s.char\Delta t(n) > 0)
              if (s.charAt(n)=='A' || s.charAt(n)=='E' || s.charAt(n)=='I' || s.charAt(n)=='O' ||
                s.charAt(n)=='U' || s.charAt(n)=='a' || s.charAt(n)=='e' || s.charAt(n)=='i' ||
                s.charAt(n)=='o' || s.charAt(n)=='u'){
                   resp= false;
              } else {
                   n++:
                   resp=isConsoante(s, n);
         } else {
                                    2º passo: Simplificação
              resp=false;
    return resp;
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n != s.length()){
        if (s.charAt(n)<'0' || s.charAt(n)>'9'){
             if (isVogal(s.charAt(n)) == true){
                 resp= false;
             } else {
                 resp=isConsoante(s, n + 1);
        } else {
             resp=false;
                                  2º passo: Simplificação
    return resp;
```

```
public static boolean isConsoante(String s, int n){
    boolean resp= true;
    if (n != s.length()){
        if (s.charAt(n)<'0' || s.charAt(n)>'9'){
             if (isVogal(s.charAt(n)) == true){
                 resp= false;
             } else {
                 resp=isConsoante(s, n + 1);
        } else {
             resp=false;
                                  O código está correto?
    return resp;
```

```
public static boolean isConsoante(String s, int i){
    boolean resp= true;

if (i == s.length()){
    resp = true;
} else if (isConsoante(s.charAt(i)) == false){
    resp = false;
} else {
    resp = isConsoante(s, i + 1);
}

return resp;
}
```

Qual das duas versões é mais fácil de entender ?

```
public static boolean isConsoante(String s, int i){
                                                       public static boolean isConsoante(String s, int i){
     boolean resp= true;
                                                             boolean resp= true;
     if (i == s.length()){}
                                                             if (i < s.length()){
          resp = true;
                                                                  if (!isConsoante(s.charAt(i))){
    } else if (isConsoante(s.charAt(i)) == false){
                                                                       resp = false;
          resp = false;
                                                                  } else {
                                                                       resp = isConsoante(s, i + 1);
    } else {
          resp = isConsoante(s, i + 1);
                                                            } else {
                                                                  resp = true;
     return resp;
                                                             return resp;
```

Qual é a sua opinião sobre o código REAL abaixo?

```
public Unidade recuperarUnidadeComCodigoDeUCI(Unidade unidadeFilha) {
    Unidade retorno = null:
    if (unidadeFilha.getCodUci() != null && !unidadeFilha.getCodUci().isEmpty()) {
        retorno = unidadeFilha;
    } else {
        retorno = unidadeFilha.getUnidadeSuperior();
    while (retorno == null || retorno.getCodUci() == null || retorno.getCodUci().isEmpty()) {
        retorno = retorno.getUnidadeSuperior();
    return retorno;
```