

```

1  program shellsort;
2
3  //Wellington de Souza Silva
4
5
6
7  //Tabela dos metodos e suas interações:
8  {
9  Interações com o metodo convencional(gap começando com a metade do vetor)
10 vetor[10.000] = 1.080.069 interações
11 vetor[1.000]  = 0.034.651 interações
12 Vetor[100]    = 0.000.525 interações
13 Vetor[10]     = 0.000.007 interações
14
15 Interações com o metodo convencional(gap começando com o tamanho do vetor)
16 vetor[10.000] = 2.146.238 interações
17 vetor[1.000]  = 0.069.074 interações
18 Vetor[100]    = 0.000.924 interações
19 Vetor[10]     = 0.000.013 interações
20
21 Interações com o metodo Knuth(gap*3+1)
22 vetor[10.000] = 0.144.339 interações
23 vetor[1.000]  = 0.008.722 interações
24 Vetor[100]    = 0.000.453 interações
25 Vetor[10]     = 0.000.013 interações
26 }
27
28
29
30
31
32
33
34 {
35 * Como escolher valores para h?
36 Existem vários modos de escolher valores para h baseados em estudos de eficiência,
37 você pode até escolher os valores que quiser, apesar de ser altamente não
38 recomendável por questões de eficiência.
39 O método de seleção de valores para h proposto por Donald Shell foi  $h = n/2$  onde n
40 é o número de elementos do vetor,
41 mas já foi provado que sua eficiência é baixa. O que vimos hoje foi o da sequência
42 de Donald Knuth, explicado como:
43  $h = 3*h+1$  E a sequência gerada: 1, 4, 13, 40, 121, 364, 1093, 3280...
44
45 Partindo de  $h = 1[*]$  precisamos do maior valor possível que seja menor que o
46 número de elementos no vetor, então basta calcular:
47  $h1 = 3*1+1 = 4$ 
48  $h2 = 3*4+1 = 13$ 
49  $h3 = 3*13+1 = 40$ 
50 * etc...
51 Para decrementar os valores de h até  $h=1$  podemos usar a fórmula inversa:
52  $h = (h-1)/3$ 
53 ...3280, 1093, 364, 121, 40, 13, 4, 1.
54 * }
55
56 const
57     MAX = 10000;
58
59 type
60     Tdado = integer;
61
62     Tvetor = array[1..MAX] of Tdado;
63
64 var
65     vetor : Tvetor;
66     contador : longint;
67
68 procedure interacoes(contador : longint);
69 var
70     arq : text;
71 begin
72     assign(arq, 'interacao.text');
73     append(arq);

```

```

70     writeln(arq, contador);
71     close(arq);
72 end;
73
74 procedure preencher(var vetor : Tvetor);
75 begin
76     vetor[1] := 4;
77     vetor[2] := 3;
78     vetor[3] := 15;
79     vetor[4] := 7;
80     vetor[5] := 10;
81     vetor[6] := 13;
82     vetor[7] := 8;
83     vetor[8] := 9;
84     vetor[9] := 2;
85     vetor[10] := 6;
86 end;
87
88 procedure manual(var vetor:Tvetor);
89 var
90     i : integer;
91 begin
92     for i := 1 to MAX do
93     begin
94         vetor[i]:=random(1500);
95     end;
96 end;
97
98 {
99     * gap intuitivo = tamanho do vetor
100    * gap tradicional(Donald Shell) = sizeVetor/2
101    * gap melhorado(Donald Knuth) = 3*gap+1;
102    * Exemplo de sequência ruim: 1, 2, 4, 8, 16,
103 }
104
105 function knuth():integer;
106 var
107     h: integer;
108 begin
109     h := 1;
110     while(h < MAX)do h := 3*h+1;
111     //writeln(h);
112     knuth := h;
113 end;
114
115 procedure exibir(vetor: Tvetor);
116 var
117     i : integer;
118 begin
119     writeln;
120     write(' | ');
121     for i:=1 to MAX do
122     begin
123         write(vetor[i]);
124         write(' | ');
125     end;
126     writeln;
127 end;
128
129 //como deslocamento
130 procedure shell knuth(var contador : longint; var vetor:Tvetor);
131 var
132     h : integer;
133     i,j : integer;
134     temp : integer;
135 begin
136     h:= knuth();
137     repeat
138     begin
139         h:=(h-1) div 3;
140
141         for i := h+1 to MAX do
142             begin

```

```

143     temp := vetor[i];
144     j := i-h;
145     while((j>=1) and (temp < vetor[j])) do
146     begin
147         vetor[j+h] := vetor[j];
148         j:=j-h;
149         contador := contador + 1;
150     end;
151     vetor[j+h] := temp;
152 end;
153 end;
154 until(h<1);
155 interacoes(contador);
156 end;
157
158 procedure shell(var contador : longint; var vetor:Tvetor);
159 var
160     gap : integer;
161     h : integer;
162     i : integer;
163     temp :integer;
164 begin
165     //gap := MAX; //nao indicado, porem intuitivo
166     gap := (MAX div 2); //gap de shell
167
168     while h>1 do
169     begin
170         for i := 1 to MAX-h+1 do
171         begin
172             if vetor[h+i-1] < vetor[i] then
173             begin
174                 contador := contador + 1;
175                 temp := vetor[i];
176                 vetor[i] := vetor[h+i-1];
177                 vetor[h+i-1] := temp;
178             end;
179             //exibir(vetor);
180         end;
181         h := h -1; //gap tradicional
182     end;
183     interacoes(contador);
184 end;
185
186 begin
187     manual(vetor);
188     exibir(vetor);
189     //shell(contador, vetor);
190     shell knuth(contador, vetor);
191     exibir(vetor);
192 end.
193

```