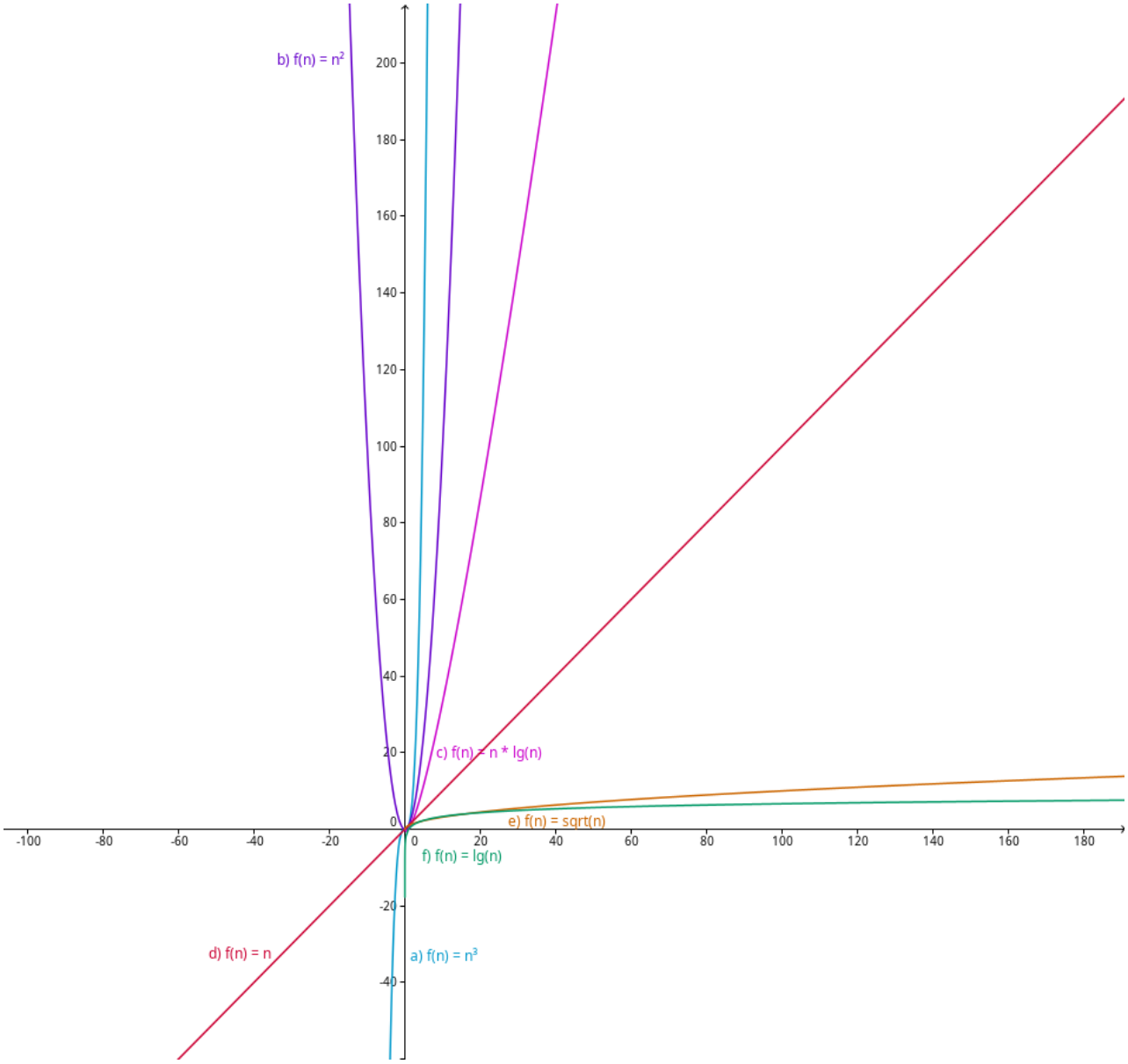
**ER 1.**
**a)** 1024
**b)** 10
**c)** ~4,087
**d)** 5
**e)** 4

**ER 2.**

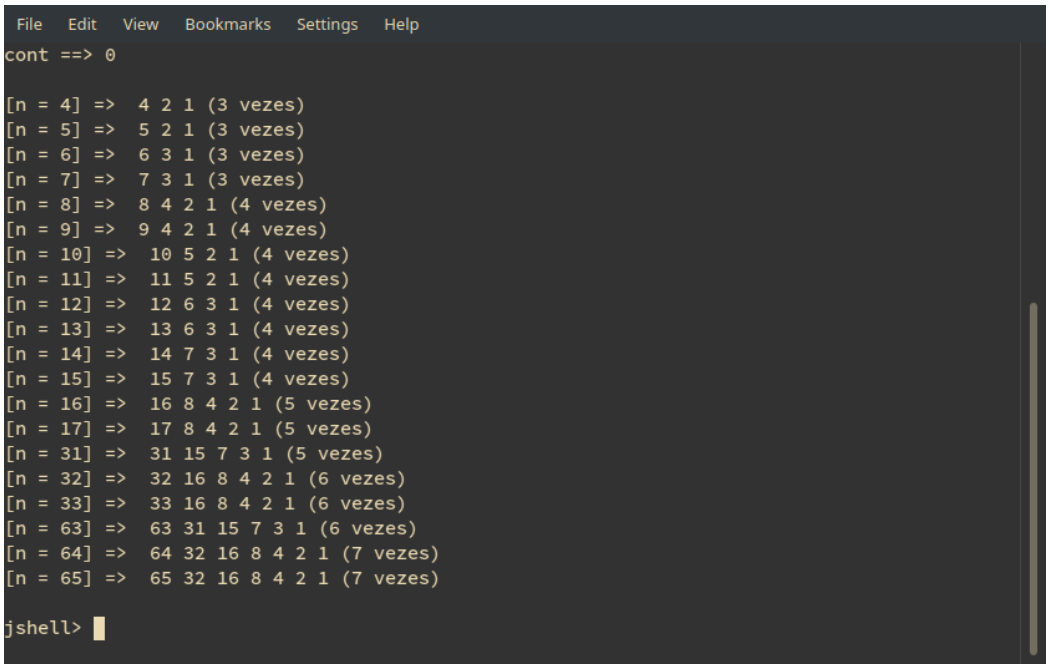



**ER 3.**
Melhor caso: $n$ subtrações – $O(n), \Omega(n), \Theta(n)$
Pior caso: $2n$ subtrações – $O(n), \Omega(n), \Theta(n)$

**ER 4.**
$n-3$ subtrações – $O(n), \Omega(n), \Theta(n)$

**ER 5.**
$lg(n)+1$ multiplicações – $O(lg(n)), \Omega(lg(n)), \Theta(lg(n))$

**ER 6.**




**ER 7.**
1) Comparação entre elementos.
2) $f(n)=n-1$ .
3) Todos os casos (pior, melhor, médio).

**ER 8.**
1) Comparação entre elementos (`min > array[i]`).
2) $f(n)=n-1$ .
3) Todos os casos (pior, melhor, médio).
4) Sim, pois é necessário comparar todos os elementos para encontrar o valor mínimo.

**ER 9.**
1) Comparação entre elementos (`array[i] == x`).
2) Pior caso: $f(n)=n$ ; Melhor caso: $f(n)=1$ ; Caso médio: $f(n)=\frac{n+1}{2}$
3) Sim, pois é necessário comparar todos os elementos para verificar se o valor procurado está no arranjo.

**E 1.**
```
int max = array[0];
int min = array[0];

for (int i = 1; i < n; i++) {
  if (array[i] > max) max = array[i];
  if (array[i] < min) min = array[i];
}
```

2 comparações realizadas n – 1 vezes → $f(n)=2(n-1)$ (pior caso, melhor caso e caso médio) → $\Theta(n)$

**E 2.**
OK

**ER 10.**
1. Custo da pesquisa sequencial: $\Theta(n)$
2. Custo da ordenação: $\Theta(n \cdot lg(n))$ + custo da pesquisa binária: $\Theta(lg(n))$

Logo, o aluno deve escolher a opção 1, por ter um custo menor.

**ER 11.**
a) F; b) V; c) V; d) V; e) V; f) F; g) F; h) V; i) F.

**E 3.**

| | O(1) | O(lg n) | O(n) | O(n.lg(n)) | $O(n^2)$ | $O(n^3)$ | $O(n^5)$ | $O(n^{20})$ |
|---|---|---|---|---|---|---|---|---|
| f(n) = lg(n) | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| f(n) = n . lg(n) | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| f(n) = 5n + 1 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $f(n) = 7n^5 - 3n^2$ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ |
| $f(n) = 99n^3 - 1000n^2$ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| $f(n) = n^5 - 99999n^4$ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |

**E 4.**

| | Ω(1) | Ω(lg n) | Ω(n) | Ω(n.lg(n)) | $\Omega(n^2)$ | $\Omega(n^3)$ | $\Omega(n^5)$ | $\Omega(n^{20})$ |
|---|---|---|---|---|---|---|---|---|
| f(n) = lg(n) | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| f(n) = n . lg(n) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| f(n) = 5n + 1 | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| $f(n) = 7n^5 - 3n^2$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| $f(n) = 99n^3 - 1000n^2$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| $f(n) = n^5 - 99999n^4$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |

**E 5.**

| | $\Theta(1)$ | $\Theta(\lg n)$ | $\Theta(n)$ | $\Theta(n.\lg(n))$ | $\Theta(n^2)$ | $\Theta(n^3)$ | $\Theta(n^5)$ | $\Theta(n^{20})$ |
|---|---|---|---|---|---|---|---|---|
| $f(n) = \lg(n)$ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| $f(n) = n \cdot \lg(n)$ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ |
| $f(n) = 5n + 1$ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| $f(n) = 7n^5 - 3n^2$ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| $f(n) = 99n^3 - 1000n^2$ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| $f(n) = n^5 - 99999n^4$ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |

**E 6.**

a) $O(n^8)$ b) $O(n^8)$ c) $O(n^3 \cdot lg(n))$ d) $O(n^8)$ e) $O(n^4 \cdot lg^3(n))$ f) $O(n^2)$

**E 7.**

a) $c=4$ e $m=6$

b) $c=1$ e $m=5$

c) Para $c<\infty$ , não há nenhum valor $c$ possível para que $c \cdot n \geq 3n^2+5n+1$ ocorra sempre a partir de um valor $m$ para infinitos $n$ .

**E 8.**

a) $c=2$ e $m=1$

b) $c=1$ e $m=1$

c) Não existe nenhuma constante positiva $c$ para que $c \cdot n^3 \leq 3n^2+5n+1$ ocorra sempre a partir de um valor $m$ para infinitos $n$ .

**E 9.**

a) $c_1=2$ e $c_2=4$

b) Para $c_2<\infty$ , não há nenhum valor $c_2$ possível para que $c_2 \cdot n \geq 3n^2+5n+1$ ocorra sempre a partir de um valor $m$ para infinitos $n$ .

c) Não existe nenhuma constante positiva $c_1$ para que $c_1 \cdot n^3 \leq 3n^2+5n+1$ ocorra sempre a partir de um valor $m$ para infinitos $n$ .

**E 10.**

OK

**ER 12.**

<u>Comparações</u>

Pior caso: $f(n)=1+2 \cdot (n-2)$ ; $O(n), \Omega(n), \Theta(n)$

Melhor caso: $f(n)=1+(n-2)$ ; $O(n), \Omega(n), \Theta(n)$

<u>Movimentações</u>

Pior caso: $f(n)=2+(n-2)$ ; $O(n), \Omega(n), \Theta(n)$

Melhor caso: $f(n)=2$ ; $O(1), \Omega(1), \Theta(1)$

**ER 13.**
Pior caso: $f(n)=n+2$ ; $O(n),\Omega(n),\Theta(n)$
Melhor caso: $f(n)=n+1$ ; $O(n),\Omega(n),\Theta(n)$

**ER 14.**
Pior caso = melhor caso: $f(n)=2n^2+n$ ; $O(n^2),\Omega(n^2),\Theta(n^2)$

**ER 15.**
Pior caso = melhor caso: $f(n)=n\cdot lg(n)+n$ ; $O(n\cdot lg(n)),\Omega(n\cdot lg(n)),\Theta(n\cdot lg(n))$

**E 11.**
Alarme
Pior caso: $f(n)=n-1$ ; $O(n),\Omega(n),\Theta(n)$
Melhor caso: $f(n)=1$ ; $O(1),\Omega(1),\Theta(1)$

Outros
Pior caso: $f(n)=2n-2$ ; $O(n),\Omega(n),\Theta(n)$
Melhor caso: $f(n)=n-1$ ; $O(n),\Omega(n),\Theta(n)$

**E 12.**
```
int max = v[0] , min = v[0];

for (int i = 1; i < n ; i++) {
    if (v[i] > max) max = v[i];
    else if (v[i] < min) min = v[i];
}
```

Pior caso: $2(n-1) = O(n)$
Melhor caso: $n-1 = O(n)$

**ER 16.**

| | Constante | Linear | Polinomial | Exponencial |
|---|---|---|---|---|
| 3n | | ✓ | | |
| 1 | ✓ | | | |
| (3/2)n | | ✓ | | |
| $2n^3$ | | | ✓ | |
| $2^n$ | | | | ✓ |
| $3n^2$ | | | ✓ | |
| 1000 | ✓ | | | |
| $(3/2)^n$ | | | | ✓ |

**ER 17.**
$f_6 < f_2 < f_1 < f_5 < f_4 < f_3$

**ER 18.**
$f_6 < f_3 < f_2 < f_9 < f_1 < f_5 < f_4 < f_7 < f_8$

**ER 19.**

| f(n) | | g(n) |
|---|---|---|
| $n + 30$ | **2** | $n^4$ |
| $n^2 + 2n - 10$ | **4** | $3n - 1$ |
| $n^3 \cdot 3n$ | **1** | $\lg(2n)$ |
| $\lg(n)$ | **3** | $n^2 + 3n$ |

**E 13.**
Neste caso, a segunda solução é mais eficiente.

**Referências**

ZIVIANI, Nivio. **Projeto de Algoritmos com implementações em Java e C++**. Cengage Learning, 2006.