

Avaliação 2 --- Segundo Bimestre

Questão 1

R1: A diferença entre um algoritmo iterativo e um recursivo está na maneira como repetem operações. Algoritmos iterativos utilizam laços (for, while) e controlam a execução com condições de parada, sendo mais eficientes em termos de memória. Já os recursivos chamam a própria função para resolver subproblemas menores, até atingir uma condição base que interrompe as chamadas. Isso consome mais memória, mas é útil para problemas naturalmente divisíveis em subproblemas semelhantes, como Fibonacci e busca em árvores.

Questão 2

b.i) $f_soma(n) = \begin{cases} 0 & \text{---} \rightarrow \text{se } n = 0 \text{ (critério de parada)} \\ 1 & \text{---} \rightarrow \text{se } n = 1 \text{ (critério de parada)} \\ n + f_soma(n-1) & \text{se } n > 1 \text{ (passo recursivo)} \end{cases}$

b.ii)

```
3  int f_soma(int n) {
4      // Critério de parada
5      if (n==0) {
6          return 0;
7      }
8      if (n==1) {
9          return 1;
10     }
11     // Passo recursivo
12     return n + f_soma(n-1);
13 }
```

c.i) $f_3em3(n) = \begin{cases} 0 & \text{---} \rightarrow \text{se } n = 0 \text{ (critério de parada)} \\ 2 & \text{---} \rightarrow \text{se } n = 1 \text{ (critério de parada)} \\ f_3em3(n-1)+3 & \text{se } n > 1 \text{ (passo recursivo)} \end{cases}$

c.ii)

```
17 int f_3em3(int n) {
18     // Critério de parada
19     if (n==0) {
20         return 0;
21     }
22     if (n==1) {
23         return 2;
24     }
25     // Passo recursivo
26     return f_3em3(n-1)+3;
27 }
```

d.i) $f_x2(n) = \begin{cases} 0 & \text{se } n = 0 \text{ (critério de parada)} \\ 3 & \text{se } n = 1 \text{ (critério de parada)} \\ f_x2(n-1)*2 & \text{se } n > 1 \text{ (passo recursivo)} \end{cases}$

d.ii)

```
29 int f_x2(int n) {
30     // Critério de parada
31     if (n==0) {
32         return 0;
33     }
34     if (n==1) {
35         return 3;
36     }
37     // Passo recursivo
38     return f_x2(n-1)*2;
39 }
```

e.i) $f_apertos(n) = \begin{cases} 0 & \text{se } n = 0 \text{ (critério de parada)} \\ 0 & \text{se } n = 1 \text{ (critério de parada)} \\ f_apertos(n-1) + (n-1) & \text{se } n > 1 \text{ (passo recursivo)} \end{cases}$

e.ii)

```
41 int f_apertos(int n) {
42     // Critério de parada
43     if (n==0) {
44         return 0;
45     }
46     if (n==1) {
47         return 0;
48     }
49     // Passo recursivo
50     return f_apertos(n-1) + (n-1);
51 }
```