

1) A função abaixo recebe uma lista encadeada com cabeça e um inteiro  $x$ , e promete devolver  $p$  tal que  $p \rightarrow \text{chave} == x$  ou NULL se tal  $p$  não existir. Analise a função verificando sua correção e eficiência.

```
struct celula {
    int         chave;
    struct celula *prox;
} ;

struct celula * busca (int x, struct celula * Y){
    int achou = 0;
    struct celula * p = Y->prox;
    while (p != NULL && !achou) {
        if (p->chave == x) achou = 1;
        p = p->prox;
    }
    if (achou) return p;
    else return NULL;
}
```

2) Prove a correção e determine o tempo de execução, no pior e no melhor caso, para o algoritmo de classificação abaixo:

```
ShakeSort(A, n)
1  e ← 1
2  para i ← n-1 até e faça
3      para j ← e até i faça
4          se A[j] > A[j+1]
5              então troca(A[j], A[j+1])
6      para j ← i até e + 1 faça
7          se A[j-1] > A[j]
8              então troca(A[j-1], A[j])
9  e ← e + 1
```

3) Projete um algoritmo, prove sua correção e determine a complexidade de tempo e comportamento assintótico, para o problema:

*Dado um inteiro  $N$ , verifique se  $N$  é primo.*

---