

1. Utilize uma das técnicas conhecidas de análise de algoritmos recursivos e forneça um limite assintótico $\theta()$ para cada algoritmo abaixo, escrito em C:

a)

```
int pow2(int a, int n) {
    if (n == 0)
        return 1;
    if (n % 2 == 0)
        return pow2(a, n/2) * pow2(a, n/2);
    else
        return pow2(a, (n-1)/2) * pow2(a, (n-1)/2) * a;
}
```

$T(n) = 2T(n/2) + 1$ portanto $\theta(n)$. A versão iterativa tem o mesmo comportamento, veja o algoritmo a seguir

```
int pow1(int a, int n) {
    int p = 1;
    for (int i = 0; i < n; i++)
        p = p * a;
    return p;
}
```

2. Apresente um programa em linguagem C, ou em pseudocódigo, para achar o valor máximo entre os n elementos de um vetor pelo método usual (iterativo) e pelo método de divisão e conquista. Para ambos os casos ache o pior caso na notação Θ . Faça uma comparação entre os dois algoritmos considerando os resultados encontrados para o pior caso na notação θ .

Versão Iterativa

```
int maxVal(int A[], int init, int end);
int max(int x, int y);
main() {
    int A[] = {100, 5, 8, 70, 2, 4, 150};
    int maximo;
    maximo = maxVal(A, 0, 6);
    printf("ValorMaximo=%d", maximo);
}

int maxVal(int A[], int init, int end) {
    int max = A[init], i;
    for(i=1; i<=end; i++) if (A[i] > max) max =
A[i];
    return max;
}
```

Como o for roda de 1 a n-1, claramente o pior caso é $\theta(n)$.

Divisão e Conquista

```
int maxVal(int A[], int init, int end);
int max(int x, int y);
main() {
    int A[] = {100, 5, 8, 70, 2, 4, 150};
    int maximo;
    maximo = maxVal(A, 0, 6);
    printf("ValorMaximo=%d", maximo);
}
```

```

}

int maxVal(int A[], int init, int end) {
    if (end - init <= 1)
        return max(A[init], A[end]);
    else {
        int m = (init + end)/2;
        int v1 = maxVal(A,init,m);
        int v2 = maxVal(A,m+1,end);
        return max(v1,v2);
    }
}

int max(int x, int y){
    if(x>y) return x;
    else return y;
}

```

$$T(n) = 2T(n/2) + C, \text{ logo } \theta(n).$$

Os dois algoritmos tem igual comportamento, ambos são $\theta(n)$. Portanto não existe melhora em se usar o método da divisão e conquista para esse caso.