SCC0220 - Laboratório Introdução à Ciência da Computação II

Relatório de execução do trabalho 2

Alunos	**NUSP**
Juan Henriques Passos	15464826
Alec Campos Aoki	15436800

Trabalho prático 2 – Exponenciação

Recursivo

Comentário

Para resolver esse problema de forma recursiva, aplicamos o algoritmo de dividir e conquistar, reduzindo o problema em problemas menores. Quando k = 1, sabemos que um número elevado a 1, é ele mesmo. Caso contrário, chamamos a função uma vez, para n elevado a k/2, e atribui esse resultado a uma variável, elevando-a ao quadrado, se k for ímpar, multiplica-se por n, para tratar o truncamento. Portando, pode-se analisar a complexidade desse algoritmo como sendo O(log k), pois a cada chamada k é dividido por dois, e isso ocorre até k ser igual a 1. Assim:
k/2^n = 1, sendo n a quantidade de vezes -> n = log k.

Código

ICMC USP SÃO CARLOS

```c
/* Implemente a exponenciação de duas formas: uma iterativa e outra utilizando o método de
 divisão e conquista (exponenciação rápida). Considere que n e k são dois inteiros, onde 0 <
 n <104 e 0 < k < 109.

Para evitar overflow, a saída deverá ser composta pelos últimos quatro dígitos do resultado.
Isso pode ser feito aplicando a operação mod (% no C) durante o cálculo. Como a operação mod
é comutativa, recomenda-se aplicá-la em cada etapa do processo.*/

#include<stdio.h>
#include<stdlib.h>
//Note que como o exericio pede apenas os 4 ultimos numeros, so precisamos operar as multiplicações com os quatro primeiros numeros.
//Dividir e conquistar.
int exp(int n,int k){
    //Pegar os primeiros 4 dígitos.
    n = n%10000;
    //Um numero elevado a 1, é ele mesmo.
    if(k == 1){
        return n%10000;
    }
    //2^8 = 2^4 * 2^4 e 2^9 = 2^4 * 2^4 * 2.
    int aux = exp(n, k/2)%10000;
    if(k%2){
        return ((aux*aux)%10000)*n;
    }
    else{
        return (aux*aux);
    }
}
```

```c
int main(){
    //n = base, k = expoente.
    int n, k, resposta;

    scanf("%d %d", &n, &k);

    resposta = exp(n, k);

    printf("%d", resposta%10000);

    return 0;
}
```

📤 Saída

```
2 64720

Recursiva
576
0.000001ms

Iterativa
576
0.000415ms
```

Iterativo

📝 Comentário

Na forma iterativa apenas aplicamos a exponenciação comum, ou seja, multiplicamos o número n, k vezes com um for, assim a complexidade é k, pois fazemos k iterações. Vale ressaltar que sempre pegamos os 4 primeiros dígitos da resposta, pois como é requisitado apenas os quatro últimos números da resposta, só é necessário os 4 primeiros algarismos dos números que serão multiplicados, dessa forma, evita-se overflow.

💻 Código

```c
#include<stdio.h>
#include<stdlib.h>

int main(){
    //n = base, k = expoente.
    int n, k, aux = 1;
    //aux
    scanf("%d %d", &n, &k);

    for(int i = 0; i<k; i++){
        aux = aux%10000;
        aux *= n;
    }

    aux = aux%10000;

    printf("%d", aux);

    return 0;
}
```

Saída

```
2 64720

Recursiva
576
0.000001ms

Iterativa
576
0.000415ms
```