DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Estrutura de Dados

Análise de Algoritmos

Professores: Luiz Chaimowicz e Raquel Prates

Análise de Algoritmos

- Determinar o **tempo de execução** de um programa pode ser um problema matemático **complexo**;

- Determinar a **ordem de complexidade** do tempo de execução, sem preocupação com o valor da constante envolvida, pode ser uma tarefa mais simples.

Análise do Tempo de Execução

- Comando de atribuição, de leitura ou de escrita: O(1).
  - `int a;`
  - `int v[15];`
  - `a = 0;`
  - `v[0] = 12;`

Análise do Tempo de Execução

- Comando de decisão:
  - Tempo dos comandos dentro do condicional, mais tempo para avaliar a condição, que é O(1).

```
if ( A[j] < A[min] )
    min = j;
```

Análise do Tempo de Execução

- Laço:
  - Soma do tempo de execução do corpo do laço **+** o tempo de avaliar a condição de parada (geralmente O(1)) **X** número de iterações.

```
sum = 0;
for (i = 0; i < sqrt(n)/2; i++) {
      if ( A[j] < A[min] )
        min = j;
     sum++;
}
```

Análise do Tempo de Execução

- Sequência de comandos:
  - Determinado pelo maior tempo de execução de qualquer comando da sequencia.

```
int sum = 0;
int i, j, k;

for (i = 0; i < sqrt(n)/2; i++)
    sum++;
for (j = 0; j < sqrt(n)/4; j++)
    sum++;
```

Análise do Tempo de Execução

- Cada chamada de uma função deve ter seu tempo computado separadamente, iniciando pelos que não chamam outros procedimentos.

- Avalia-se então os chamam os já avaliados (utilizando os tempos desses).

- O processo é repetido até chegar no programa principal (função main)

Análise do Tempo de Execução

- Qual a **função de complexidade** f(n) para o número de atribuições à variável a?

```
int exemplo1(int n){
     int i, a;
     a=0;
     for (i = 0; i < n; i++)
          a += i;
     return a;
}
```

Análise do Tempo de Execução

- Qual a **função de complexidade** g(n) para o número de atribuições à variável a?

```
void exemplo2 (int n){
     int i, j, a;
     a = 0;
     for (i = 0; i < n; i++)
          for (j = n; j > i; j--)
               a += i+j
     a = exemplo1(a);
}
```

Análise do Tempo de Execução

- O que faz essa função? Qual a sua ordem de complexidade?

```
void func(int *A, int *B, int *C, int n){
     int i, j, k;

     for(i=0; i<n; i++)
       for(j=0; j<n; j++){
          C[i][j]=0;
          for(k=n-1; k>=0; k--)
             C[i][j]=C[i][j]+A[i][k]*B[k][j];
     }
}
```

Exemplo: Algoritmo de Ordenação

Algoritmo da Seleção

- Seleciona o menor elemento do conjunto.
- Troca este com o primeiro elemento A[0].
- Repita as duas operações acima com os *n* - 1 elementos restantes, depois com os *n* - 2, até que reste apenas um.

```
void Ordena(Vetor A) {
 /*ordena o vetor A em ordem ascendente*/
   int i, j, min, x;
   for (i = 0; i < n-1; i++) {
      min = i;
      for (j = i + 1; j < n; j++)
          if ( A[j] < A[min] )
              min = j;
      /*troca A[min] e A[i]*/
      x = A[min];
      A[min] = A[i];
      A[i] = x;
   }
}
```