Vinicius Ramos Deramio (bv3015823)

Análise de Finitude, Corretude e Complexidade de Tempo de um algoritmo

Algoritmo analisado:

```
61  int buscaPosicaoArray(int *array, int tamanhoArray, int chave){
62
63      for(int i=0; i<tamanhoArray; i++){
64          if(array[i] == chave){
65              return i+1;
66          }
67      }
68
69      return NULL;
```

Análise de Finitude:

Para analisarmos a finitude de um algoritmo devemos determinar se ele é finito através da análise de suas estruturas de repetições , nesta situação analisaremos se existe algum caso em que a estrutura de repetição “for” seja infinita. No algoritmo em questão, a variável “i” é inicializada, verificada se é menor que o tamanho do array e depois é incrementado 1 em seu valor. Nota-se que a condição de existência da estrutura é se “i” é menor que o tamanho do array, assim quando “i” for maior que esse valor a estrutura se encerrará, portanto conclui-se que o algoritmo é finito.

Análise de Corretude:

Primeiro é necessário determinar a invariante principal do algoritmo, neste caso: o laço de repetição retorna a posição encontrada no array caso o número seja igual a chave procurada, caso contrário ele só partirá para a próxima iteração.

Pontos críticos da análise:

- Na primeira iteração, onde é i = 0, se o array[0] possuir a chave, número procurado, o laço se encerra e retorna i +1, senão continua para a próxima iteração.
- Em alguma iteração aleatória, se o array[i] possuir a chave o laço se encerra e retorna i +1, senão continua para a próxima iteração.

- Na última iteração, se o array[i] possuir a chave o laço se encerra e retorna i +1, senão o algoritmo retorna NULL provando que a chave não pode ser encontrada.

Análise de Complexidade de Tempo:

Deve ser analisado o melhor caso e o pior caso:

Melhor caso:

Linha	Código	Custo	# execuções
63	for(int i=0; i<tamanhoArray;i++){	c1	1
64	if(array[i] == chave){	c2	1
65	return i+1	c3	1
66	}	0	0
67	}	0	0
68		0	0
69	return NULL	c7	0

$$T(n) = c1 + c2 + c3$$

Pior caso:

Linha	Código	Custo	# execuções
63	for(int i=0; i<tamanhoArray;i++){	c1	n
64	if(array[i] == chave){	c2	n-1
65	return i+1	c3	0
66	}	0	n-1
67	}	0	n-1
68		0	n-1
69	return NULL	c7	1

$$T(n) = (c1 + c2)*n + c6 - 1$$