Aula 10 – merge correção- gabarito

quinta-feira, 17 de fevereiro de 2022 20:47

Considere o algoritmo merge a seguir:

Algorithm 2: merge(A, p, q, r)

```
1  n1 = q − p + 1 ;
2  n2 = r − q ;
3  let L[1..n1 + 1] and R[1..n2 + 1] be new arrays;
4  for i = 1 to n1 do
5  |  L[i] = A[p + i − 1];
6  end
7  for j = 1 to n2 do
8  |  R[j] = A[q + j];
9  end
10 L[n1 + 1] = ∞;
11 R[n2 + 1] = ∞;
12 i = 1;
13 j = 1;
14 for k = p to r do
15 |  if L[i] ≤ R[j] then
16 |  |  A[k] = L[i];
17 |  |  i = i + 1;
18 |  end
19 |  else
20 |  |  A[k] = R[j];
21 |  |  j = j + 1;
22 |  end
23 end
```

Prove a seguinte invariante de laço e conclua que o algoritmo merge *é correto:*

> *Antes de cada iteração do* **for** *(linhas 14-23), o subvetor* $A[p..k-1]$ *está ordenado, e contém os* $k-p$ *menores elementos dos vetores* $L[1..n_1+1]$ *e* $R[1..n_2+1]$.

Inicialização: Antes da primeira iteração do laço FOR temos que $K = p$, e portanto o vetor $A[p..p-1]$ é o vetor vazio. Assim, a invariante acima é trivialmente verdadeira neste caso.

Manutenção: Considere uma iteração arbitrária onde $p < K \leq r$. Estamos supondo que a invariante é verdadeira antes desta iteração. Observe que a cada iteração i ou j é incrementado de acordo com o resultado da comparação $L[i] \leq R[i]$. Temos, assim, 2 casos a considerar:

① $L[i] \leq R[j]$: Neste caso, o elemento $L[i]$ é o menor dos elementos dos vetores $L[1..n_1+1]$ e $R[1..n_2+1]$ que ainda não foram copiados para A. Como, por hipótese o vetor $A[p..K-1]$ está ordenado e possui os $(K-p)$ menores elementos dos vetores $L[1..n_1+1]$ e $R[1..n_2+1]$, concluímos que $A[p..K]$ está ordenado e possui os $(K+1-p)$ menores elementos dos vetores $L[1..n_1+1]$ e $R[1..n_2+1]$ após a execução da linha 16 que copia $L[i]$ para a posição K do vetor A.

② $L[i] > R[j]$: A argumentação é análoga considerando que $R[j]$ é o menor dos elementos dos vetores $L[1..n_1+1]$ e $R[1..n_2+1]$.

Finalização: Ao término da execução do laço FOR, temos que $K = r+1$, e portanto a invariante nos dá que o subvetor $A[p..r]$ está ordenado e possui os $(r-p+1) = n_1+n_2$ menores elementos dos vetores $L[1..n_1+1]$ e $R[1..n_2+1]$. Como todos os elementos dos vetores $L[1..n_1+1]$ e $R[1..n_2+1]$ foram copiados para A, concluímos que o algoritmo merge é correto. ▨