

3ª Avaliação - CANA - Assinlerena

1-a)

1. Procedimento caminhos Curtos (Grafo, origem) {
2. Para (cada Vértice V no Grafo) {
3. $V.dist \leftarrow \infty$;
4. $V.anterior \leftarrow \text{Nulo}$;
5. }
6. origem.dist $\leftarrow 0$;
7. $C \leftarrow$ conjunto de todos os vértices do Grafo;
8. Enquanto (tiver elementos em C) {
9. $M \leftarrow$ Vértice em C com menor .dist;
10. remove M de C ;
11. Para (cada vizinha V de M) {
12. $aux \leftarrow M.dist + \text{distancia}(M, V)$;
13. Se ($aux < V.dist$ e $\text{calcNumVerticesNE}(\text{Grafo}, \text{origem}, V) \leq 1$) {
14. $V.dist \leftarrow aux$;
15. $V.anterior \leftarrow M$;
16. }
17. }
18. }
19. retorna Grafo; // os caminhos são salvos em $V.anterior$.
20. }

CONTINUA

```

1. Algoritmo calcNumVerticesNE(Grafo, origem, V) {
2.   int numVerticesNE ← 0;
3.   Enquanto (V ≠ origem) {
4.     Se (V não Existe) {
5.       numVerticesNE++;
6.     }
7.     V ← V. anterior;
8.   }
9.   retorna numVerticesNE;
10.}

```

b) Algoritmo calcNumVerticesNE

linhas 4-9: $\Theta(|V|) \rightarrow$ tempo total $\Theta(|V|)$.

Procedimento caminhos Curtos

linhas 2-5: $\Theta(|V|)$

linha 6: $\Theta(1)$

linha 7: $\Theta(V)$

linha 8: $\Theta(|V|)$, linha 11: $\Theta(|V|)$, linha 13: $\Theta(|V|) \Rightarrow \Theta(|V|^3)$

tempo total = $\Theta(|V|^3)$.

2-a)

Procedimento ocupaLaboratorio($A[1..n]$) { // A = AtividadesPara (i de 1 até n) { $i.prioridadeSelecao \leftarrow (i.duracao / numConflitos(i));$

}

 $mergeSort(A[1..n]);$ // ordem de .prioridadeSelecao $A[1..n] \leftarrow A[n..1]$ // inverte vetor $solucao \leftarrow [A[1]]$ Para (j de 2 até n) { $adiciona \leftarrow true;$ Para (k de 1 até $tam(solucão)$) {Se (j conflita com k) { // se atividade j conflita com solução $adiciona \leftarrow false;$

}

}

Se ($adiciona == true$) { $solucao.addFinal(j);$ // atividade j .

}

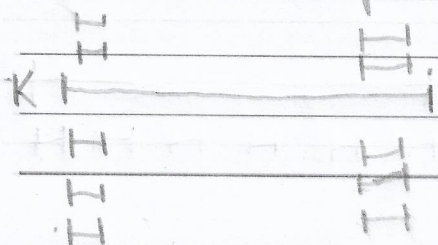
}

Retorna $solucao;$

}

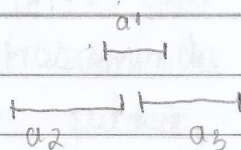
Obs: A escolha gulosa é sempre a atividade que possui maior prioridade ($duracao / numero\ de\ conflitos$) e que não conflita com outra atividade já na solução.

b) Contra exemplo:



! O grande número de conflitos afeta a prioridade de seleção da atividade ótima K .

3- a)



No contra exemplo acima, o algoritmo selecionaria a atividade a_1 e eliminaria as atividades a_2 e a_3 . Mas a solução ótima seria as atividades a_2 e a_3 .

c)

Procedimento $SeAtiv(A[1..n]) \{$

$Se (n=0)$ retorna (vazio)

Se (primeira chamada ao procedimento) $\{$

$a-gul \leftarrow$ atividade com menos conflitos em A

$\}$ Senão $\{$

$a-gul \leftarrow$ atividade que termina mais cedo em A

$\}$

$A' \leftarrow$ Elimino-conflitos($A, a-gul$)

$S \leftarrow$ SeAtividade-gul(A')

$S \leftarrow S + a-gul$

retorna(S)

$\}$

d) A única diferença entre esse problema e o da aula 15 é saber por onde começar. Podemos escolher a atividade com menos conflitos como ponto de partida e então resolver o mesmo problema se torna do mesmo tipo do da aula 15, para o qual já temos uma solução gulosa ótima.

Declare não trazer informações da prova com colegas e
nem pesquisar na internet.

José Douglas Gondim Soares.

CONTINUA