

Pré-visualizar teste: Exame

Conteúdo

Informações do teste Descrição Prova individual sem consulta. Qualquer evidência do contrário nas respostas dadas implicará a reprovação à disciplina e demais consequências previstas no regulamento da Universidade. Instruções Teste Este teste tem um limite de tempo de 2 horas e 30 minutos. Este teste será salvo e enviado cronometrado automaticamente quando o prazo expirar.

temporizador não aparece ao fazer uma visualização prévia deste Testel

Não permitido. Este teste só pode ser feito uma vez.

Avisos aparecem quando restar metade do tempo, 5 minutos, 1 minuto e 30 segundos./O

¥ Estado de Conclusão da Pergunta:

Várias

conclusao timer continuara a contar se voce sair do teste.

PERGUNTA 1

2 pontos

Salvar resposta

Considere o problema de, num array de inteiros, determinar o comprimento do maior subarray sem elementos repetidos.

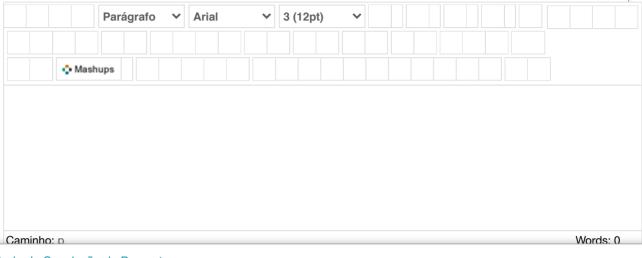
Para isso definiu-se uma função auxiliar que calcula o tamanho do maior sufixo sem elementos repetidos.

```
int auxiliar (int v[], int N) {
    int i, j, found=0;

// Pre: N>0
    i=N-2; found = 0;
    while (i>=0 && !found) {
        // Inv: ???
        j=N-1;
        while (v[j] != v[i]) j--;
        if (j==i) i--; else found = 1;
    }
    // Pós: -1<=i<=N-2 && nrep (v,i+1,N) && (i<0 || exists (i<k<N) v[k] == v[i])
    return (N-i-1);
}</pre>
```

```
if (t>r) r = t;
return r;
}
```

Apresente um invariante para o ciclo exterior da função auxiliar que permita verificar a respectiva pós-condição. O predicado nrep(v,i,j) testa que não existem repetidos no array v entre as posições i e j-1.



▼ Estado de Conclusão da Pergunta:

PERGUNTA 2

De forma a analisar a complexidade da função maiorsubseq, identifique o melhor e pior casos da execução da função auxiliar definida acima, considerando o número de comparações efectuadas entre elementos do array. Para o pior caso diga também, justificando, qual o número de comparações efectuadas.



PERGUNTA 3

2 pontos Salvar resposta

2 pontos

Salvar resposta

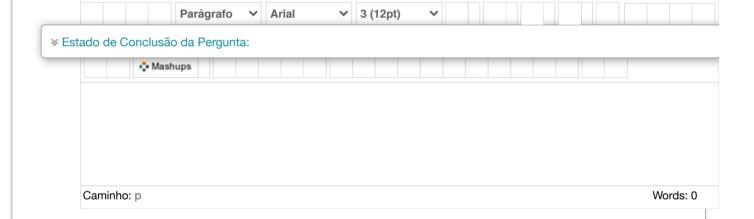
O pior caso da execução de maiorsubseq corresponde ao pior caso da execução da função auxiliar. Para este caso diga, justificando, qual o número de comparações efectuadas pela função maiorsubseq entre elementos do array. Para isso, apresente e resolva uma recorrência que traduza esse número em função do tamanho do array.

Caminho: p Words: 0

PERGUNTA 4

2 pontos Salvar resposta

Tendo em consideração que o pior caso da execução de maiorsubseq corresponde ao pior caso da execução da função auxiliar, apresente uma definição optimizada da função maiorsubseq que tira partido desse facto, e em que o tempo de execução nesse caso é substancialmente reduzido. Identifique ainda, justificando, o pior caso de execução dessa versão optimizada.



PERGUNTA 5

2 pontos Salvar resposta

Uma forma de optimizar este algoritmo passa por calcular, para cada posição k do array, a última posição (entre 0 e k - 1) onde esse elemento já apareceu no array (-1 caso seja a primeira ocorrência). A seguinte função faz esse cálculo de uma forma muito pouco eficiente (O (N^2)).

```
void ultimo (int v[], int N, int ult[]) {
  int i,j;
  for (i=0;i<N;i++)
    for (j=i-1; j>=0 && v[i] != v[j]; j--);
    ult[i] = j;
}
```

Uma forma alternativa de preencher o array ult consiste em usar uma tabela de Hash onde se percorre o array v da esquerda para a direita e se guarda, para cada elemento a última posição onde ele correu. Mostre como poderia declarar a tabela de Hash que seria necessária para implementar esta versão optimizada da função ultimo, por forma a garantir que o tempo

ant contem uma floresta de arvores de travessia.

```
int travessiaBFTreeComplete(Graph g, int ant[]) {
  int i;
  for(i=0; i<NV; i++) ant[i] = -2;
  for(i=0; i<NV; i++) if (ant[i] == -2) travessiaBFTree(g,i,ant);
}</pre>
```

Considere agora uma execução desta função sobre um grafo não orientado com 6 vértices, que produziu o seguinte vector ant [-1,-1,0,4,5,0].

Indique se é possível usando este vector responder a cada uma das seguintes questões, e em caso afirmativo diga qual é a resposta respectiva:

- 1. Os vértices 2 e 3 pertencem ao mesmo componente ligado do grafo?
- 2. Qual é a distância (comprimento do caminho mais curto) entre os vértices 2 e 3?
- 3. Qual é a distância entre os vértices 4 e 0?



¥ Estado de Conclusão da Pergunta:

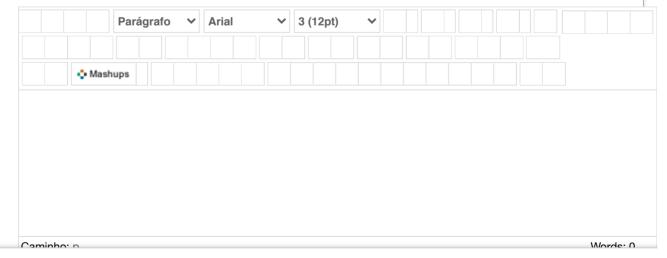
PERGUNTA 9

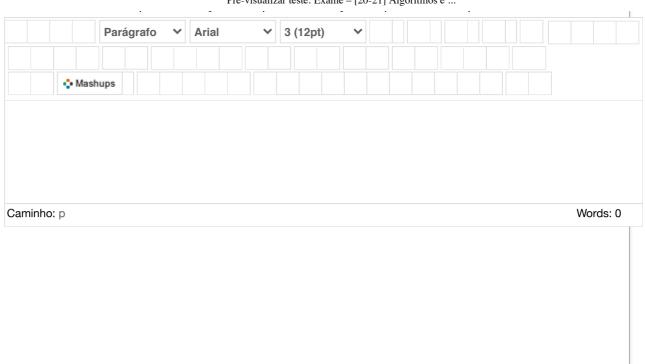
2 pontos Salvar resposta

Complete a definição da seguinte função, que "pinta" os vértices de um grafo não orientado com o componente a que pertencem, i.e., preenche o vector comp colocando em cada posição u um número inteiro (0, 1, 2, ...) que identifica o componente a que pertence o vértice u. A função começa por chamar a

função travessiaBFTreeComplete, trabalhando depois a partir do vector ant (note que dois vértices pertencem ao mesmo componente se fizerem parte da mesma árvore de travessia).

```
int componentes (Graph g, int comp[]) {
  int ant[NV];
  travessiaBFTreeComplete(g, ant);
  // Completar
}
```





▼ Estado de Conclusão da Pergunta:

 ${\it Clique\ em\ Salvar\ e\ Enviar\ para\ salvar\ e\ enviar.\ Clique\ em\ Salvar\ todas\ as\ respostas}$